MARIANA RAMOS FRANCO

**AVALIAÇÃO ORGANIZACIONAL DE TIMES DE
AGENTES PARA O *MULTI-AGENT PROGRAMMING*
*CONTEST***

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Ciências.

São Paulo
2014

MARIANA RAMOS FRANCO

**AVALIAÇÃO ORGANIZACIONAL DE TIMES DE
AGENTES PARA O *MULTI-AGENT PROGRAMMING*
*CONTEST***

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Ciências.

Área de Concentração:

Engenharia de Computação

Orientador:

Prof. Dr. Jaime Simão Sichman

São Paulo
2014

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 01 de julho de 2014.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Franco, Mariana Ramos

Avaliação Organizacional de Times de Agentes para o *Multi-Agent Programming Contest* / M. R. Franco. – versão corr. – São Paulo, 2014.

114 p.

Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Sistemas multiagentes 2.Avaliação organizacional em sistemas multiagentes I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. t.

AGRADECIMENTOS

Agradeço primeiramente ao meu orientador, Prof. Dr. Jaime Simão Sichman, pelo incentivo, orientação, disponibilidade, paciência e, principalmente, pelas sugestões dadas durante todo o Mestrado.

Aos meus colegas e professores do Laboratório de Técnicas Inteligentes (LTI), pelo apoio e discussões sobre este trabalho, e também pelos momentos de descontração.

Agradeço à Maisa, minha gerente na IBM, por não poupar esforços para flexibilizar minha rotina nos momentos mais difíceis.

Ao meu namorado Flavio, pelo carinho, amizade, paciência e incondicional apoio.

Aos meus pais, Edson e Waldinéia, por sempre me incentivarem, e pela compreensão nos momentos em que precisei estar ausente. Agradeço também aos dois “malinhas”, meus melhores amigos e irmãos, Vinicius e Fernando, pela amizade e carinho que nos une.

Por fim, gostaria ainda de agradecer a todos meus amigos, que próximos ou à distância, foram de extrema importância para que eu atingisse mais este objetivo.

RESUMO

Um subconjunto importante da pesquisa em sistemas multiagentes (SMA) baseia-se no estudo das organizações. A organização define a estrutura do SMA e as regras que os agentes devem seguir, a fim de aumentar a eficiência do sistema. No entanto, dado um domínio, a escolha da organização que melhor resolve o problema ainda é uma questão sem resposta. Assim, abordagens empíricas para a avaliação de organizações são importantes, pois fornecem indícios valiosos sobre os custos e benefícios de diferentes configurações organizacionais, ajudando desenvolvedores e projetistas na definição da organização a ser adotada. Neste contexto, este trabalho, compara e avalia o impacto da mudança de parâmetros organizacionais no desempenho de um SMA, cujo objetivo é competir no cenário “Agents on Mars” proposto no *Multi-Agent Programming Contest* (MAPC).

ABSTRACT

An important subset of multi-agent systems (MAS) are based on the study of organizations. The organization defines the MAS structure and the rules which the agents must follow, increasing the MAS efficiency. Given an application domain, however, the choice of a particular organization that better solves the problem is still an open problem. Therefore, empirical approaches to the evaluation of organizations are important since they provide valuable evidences about the costs and benefits of different organizational settings, helping developers and designers to define the organization to be adopted. In this context, this work compares and evaluates the impact of organizational changes in the performance of a MAS, whose goal is to evolve in the “Agents on Mars” scenario proposed in the Multi-Agent Programming Contest (MAPC).

SUMÁRIO

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

1	Introdução	14
1.1	Motivação	15
1.2	Objetivos	15
1.3	Metodologia	15
1.4	Organização do Texto	17
2	Sistemas Multiagentes	18
2.1	Conceitos Básicos	18
2.1.1	Agentes	19
2.1.2	Ambientes	22
2.1.3	Interações	24
2.1.4	Organizações	25
2.2	Organizações em Sistemas Multiagentes	25
2.2.1	Origem e Visibilidade das Organizações	25
2.2.2	SMA Centrados nas Organizações	27
2.2.3	Paradigmas Organizacionais	27
2.3	Programação de Sistemas Multiagentes	30
2.3.1	Programação de Agentes	31
2.3.2	Programação de Ambientes	35

2.3.3	Programação de Interações	37
2.3.4	Programação de Organizações	38
2.3.5	Arcabouço JaCaMo	40
2.4	Considerações Finais	42
3	<i>Multi-Agent Programming Contest</i>	43
3.1	Cenário <i>Agents on Mars</i>	43
3.2	<i>LTI-USP Team</i>	47
3.2.1	Arquitetura	47
3.2.2	Estratégias	49
3.3	Resultados	55
3.3.1	MAPC 2012	55
3.3.2	MAPC 2013	56
3.4	Considerações Finais	57
4	Avaliação Organizacional em SMA	58
4.1	Avaliação Organizacional	58
4.2	Métricas	59
4.3	Testes Estatísticos	60
4.4	Metodologia	64
4.5	Considerações Finais	64
5	Experimentos e Resultados	65
5.1	Experimentos	65
5.1.1	Variação de Parâmetros Organizacionais	66
5.1.2	Características do Ambiente	68
5.1.3	Execução dos Experimentos	70

5.2	Metodologia de Análise	71
5.3	Análise dos Resultados	74
5.3.1	Efeito do tamanho dos <i>clusters</i>	74
5.3.2	Efeito da homogeneidade dos <i>clusters</i>	77
5.3.3	Efeito da dispersão dos <i>clusters</i>	80
5.3.4	Efeito do número de <i>clusters</i>	83
5.4	Considerações Finais	85
6	Conclusões	86
	Referências	88
	Apêndice A - Resultados dos Experimentos	93
A.1	Cenário SC1_P	93
A.2	Cenário SC1_G	95
A.3	Cenário SC2_SN	96
A.4	Cenário SC2_SS	98
A.5	Cenário SC2_NS	99
A.6	Cenário SC3_SS	101
A.7	Cenário SC3_SN	102
A.8	Cenário SC4_SN	104
A.9	Cenário SC4_SS	105
A.10	Cenário SC4_NN	107
A.11	Cenário SC5_NN	108
A.12	Cenário SC5_SS	110
A.13	Cenário SC6_SS	111
A.14	Cenário SC6_SN	113

LISTA DE FIGURAS

1	Estrutura de um Sistema Multiagente (JENNINGS; WOOLDRIDGE, 2001).	19
2	Agentes interagem com o ambiente através de sensores e atuadores (RUSSELL; NORVIG, 2010).	20
3	Agente reativo (RUSSELL; NORVIG, 2010).	20
4	Agente cognitivo (DEMAZEAU; MÜLLER, 1990).	21
5	Agente BDI.	22
6	Exemplo de hierarquia. (HORLING; LESSER, 2004)	28
7	Exemplo de coalizões. (HORLING; LESSER, 2004)	29
8	Exemplo de mercado. Os compradores são representados em branco e os vendedores em preto. (HORLING; LESSER, 2004)	29
9	Exemplo de time. (HORLING; LESSER, 2004)	30
10	Representação dos <i>workspaces</i> (RICCI; VIROLI; OMICINI, 2006).	36
11	Representação de um artefato (RICCI; PIUNTI; VIROLI, 2010).	36
12	Visão geral de um SMA <i>JaCaMo</i> , destacando suas três dimensões (BOISSIER et al., 2011).	42
13	Cenário “ <i>Agents on Mars</i> ”.	45
14	Arquitetura do “ <i>LTI-USP Team</i> ”.	48
15	Especificação Estrutural do “ <i>LTI-USP Team</i> ” 2012.	50
16	Especificação Funcional do “ <i>LTI-USP Team</i> ” 2012.	51
17	Especificação Estrutural do “ <i>LTI-USP Team</i> ” 2013.	53
18	Especificação estrutural dos times.	66
19	Cenários.	69
20	Cenários (continuação).	70

21	Cenário SC3_SS, contendo 3 <i>clusters</i>	72
22	Cenário SC3_SS - Resultados das simulações TG2 vs TG4.	72
23	Cenário SC3_SS - Grafo de ordem parcial dos times.	73
24	SC1_P vs SC1_G - Mapas.	74
25	SC1_P vs SC1_G - Grafos de ordem parcial.	75
26	SC1_P vs SC1_G - TG1 (verde) vs TG10 (azul)	75
27	SC1_P vs SC1_G - TG6 (azul) vs TG3 (verde)	76
28	SC1_P vs SC1_G - Divisão dos <i>clusters</i> em zonas.	76
29	SC2_SS vs SC2_NS - Mapas.	77
30	SC2_SS vs SC2_NS - Grafos de ordem parcial.	77
31	SC2_SS vs SC2_NS - TG8 (azul) vs TG3 (verde)	78
32	SC2_SS vs SC2_NS - TG6 (azul) vs TG2 (verde)	79
33	SC4_SN vs SC4_NN - Mapas.	79
34	SC4_SN vs SC4_NN - Grafos de ordem parcial.	79
35	SC4_SN vs SC4_NN - TG3 (azul) vs TG2 (verde)	80
36	SC2_SN vs SC2_SS - Mapas.	81
37	SC2_SN vs SC2_SS - Grafos de ordem parcial.	81
38	SC2_SN vs SC2_SS - TG3 (verde) vs TG2 (azul)	82
39	SC6_SS vs SC6_SN - Mapas.	82
40	SC6_SS vs SC6_SN - Grafos de ordem parcial.	82
41	SC6_SS vs SC6_SN - TG4 (verde) vs TG8 (azul)	83
42	SC3_SS vs SC4_SS vs SC5_SS - Mapas.	84
43	SC3_SS vs SC4_SS vs SC5_SS - Grafos de ordem parcial.	84
44	Cenário SC1_P - Mapa.	93
45	Cenário SC1_P - Grafo de ordem parcial.	93
46	Cenário SC1_G - Mapa.	95

47	Cenário SC1_G - Grafo de ordem parcial.	96
48	Cenário SC2_SN - Mapa.	96
49	Cenário SC2_SN - Grafo de ordem parcial.	97
50	Cenário SC2_SS - Mapa.	98
51	Cenário SC2_SS - Grafo de ordem parcial.	99
52	Cenário SC2_NS - Mapa.	99
53	Cenário SC2_NS - Grafo de ordem parcial.	100
54	Cenário SC3_SS - Mapa.	101
55	Cenário SC3_SS - Grafo de ordem parcial.	102
56	Cenário SC3_SN - Mapa.	102
57	Cenário SC3_SN - Grafo de ordem parcial.	103
58	Cenário SC4_SN - Mapa.	104
59	Cenário SC4_SN - Grafo de ordem parcial.	105
60	Cenário SC4_SS - Mapa.	105
61	Cenário SC4_SS - Grafo de ordem parcial.	106
62	Cenário SC4_NN - Mapa.	107
63	Cenário SC4_NN - Grafo de ordem parcial.	107
64	Cenário SC5_NN - Mapa.	108
65	Cenário SC5_NN - Grafo de ordem parcial.	108
66	Cenário SC5_SS - Mapa.	110
67	Cenário SC5_SS - Grafo de ordem parcial.	111
68	Cenário SC6_SS - Mapa.	111
69	Cenário SC6_SS - Grafo de ordem parcial.	112
70	Cenário SC6_SN - Mapa.	113
71	Cenário SC6_SN - Grafo de ordem parcial.	114

LISTA DE TABELAS

1	Tipos de Eventos	34
2	Papéis e ações.	46
4	Especificação Deontica do “ <i>LTI-USP Team</i> ” 2012.	52
5	Resultado final do MAPC 2012.	55
6	Resultado final do MAPC 2013.	57
7	Testes estatísticos (TRIOLA, 2008).	62
8	Divisão dos agentes em pelotões.	67
9	Características dos cenários.	69
10	Cenário SC3_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 73	
11	Cenário SC1_P - Média e desvio padrão dos confrontos.	94
12	Cenário SC1_P - Resultado do teste de “Wilcoxon Pareado” (valor-p). 94	
13	Cenário SC1_G - Média e desvio padrão dos confrontos.	95
14	Cenário SC1_G - Resultado do teste de “Wilcoxon Pareado” (valor-p). 96	
15	Cenário SC2_SN - Média e desvio padrão dos confrontos.	97
16	Cenário SC2_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 97	
17	Cenário SC2_SS - Média e desvio padrão dos confrontos.	98
18	Cenário SC2_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 99	
19	Cenário SC2_NS - Média e desvio padrão dos confrontos.	100
20	Cenário SC2_NS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 100	
21	Cenário SC3_SS - Média e desvio padrão dos confrontos.	101
22	Cenário SC3_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 102	
23	Cenário SC3_SN - Média e desvio padrão dos confrontos.	103
24	Cenário SC3_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 103	

25	Cenário SC4_SN - Média e desvio padrão dos confrontos.	104
26	Cenário SC4_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 105	
27	Cenário SC4_SS - Média e desvio padrão dos confrontos.	106
28	Cenário SC4_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 106	
29	Cenário SC4_NN - Média e desvio padrão dos confrontos.	107
30	Cenário SC4_NN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 108	
31	Cenário SC5_NN - Média e desvio padrão dos confrontos.	109
32	Cenário SC5_NN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 109	
33	Cenário SC5_SS - Média e desvio padrão dos confrontos.	110
34	Cenário SC5_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 111	
35	Cenário SC6_SS - Média e desvio padrão dos confrontos.	112
36	Cenário SC6_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p). 112	
37	Cenário SC6_SN - Média e desvio padrão dos confrontos.	113
38	Cenário SC6_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p). 114	

LISTA DE ABREVIATURAS E SIGLAS

ACMAS	<i>Agent-Centered Multi-Agent Systems</i>
AEIO	<i>Agent, Environment, Interaction, Organization</i>
API	<i>Application Programming Interface</i>
A&A	Agentes e Artefatos
BDI	<i>Beliefs-Desires-Intentions</i>
COT	<i>Computation Organization Theory</i>
ED	Especificação Deôntica
EE	Especificação Estrutural
EF	Especificação Funcional
EnO	Entidade Organizacional
EO	Especificação Organizacional
IO	Infraestrutura Organizacional
LMO	Linguagem de Modelagem de Organizações
LTI	Laboratório de Técnicas Inteligentes
MAPC	<i>Multi-Agent Programming Contest</i>
OCMAS	<i>Organization-Centered Multi-Agent Systems</i>
SCH	Esquema Social
SMA	Sistemas Multiagentes
USP	Universidade de São Paulo

1 INTRODUÇÃO

Ao longo dos últimos anos, graças à evolução e gradual amadurecimento das tecnologias para o desenvolvimento de Sistemas Multiagentes (SMA), estes vêm sendo cada vez mais utilizados como solução viável para a concepção e implementação de sistemas abertos e distribuídos.

Essa maior adoção tem feito com que propriedades não-funcionais dos SMA, tais como desempenho, escalabilidade e estabilidade, que vinham até então recebendo pouca atenção dos pesquisadores, ganhassem cada vez mais importância na definição e projeto de tais sistemas.

Prever o desempenho de um SMA, em particular, é uma questão bastante difícil já que várias são as variáveis que podem influenciar o seu funcionamento, tais como o número de tarefas concorrentes, características do ambiente, o número de agentes e a sua organização.

A organização é um componente fundamental para restringir o comportamento dos agentes, a fim de aumentar a eficiência global do sistema. Entretanto, ao mesmo tempo, a organização pode afetar de forma adversa o desempenho do SMA, aumentando a carga de comunicação, reduzindo sua flexibilidade, e/ou adicionando uma camada adicional de complexidade ao sistema.

Desta forma, este trabalho faz uma avaliação do impacto da mudança de certos parâmetros organizacionais no desempenho de um SMA. Para isso foram analisados dados empíricos de diferentes times de agentes que competem no *Multi-Agent Programming Contest* (MAPC), uma competição internacional para comparar e avaliar técnicas de programação de sistemas multiagentes.

1.1 Motivação

Em SMA abertos e com um número elevado de agentes, a organização é um componente fundamental para restringir o comportamento dos agentes, a fim de aumentar a eficiência global do sistema e diminuir a sua complexidade. Entretanto, a organização dos agentes é um dos fatores que podem influenciar em muito o desempenho do SMA. Desta forma, um dos principais problemas na área de SMA é determinar que organização adotar para cada tipo de problema.

No entanto, dado um domínio, a escolha da organização que melhor resolve o problema ainda é uma questão sem resposta. Assim, abordagens empíricas para a avaliação de organizações são importantes, pois fornecem indícios valiosos sobre os custos e benefícios de diferentes configurações organizacionais, ajudando desenvolvedores e projetistas na definição da organização a ser adotada.

1.2 Objetivos

O objetivo deste trabalho é demonstrar, de modo empírico, como mudanças em alguns parâmetros organizacionais podem afetar o desempenho de um SMA num cenário específico.

O parâmetro organizacional tratado neste trabalho é o número de grupos de agentes que devem resolver um problema em uma determinada organização. Em particular, será estudado como a alteração deste número pode influenciar o desempenho de um time de agentes no *Multi-Agent Programming Contest* (MAPC). Além deste parâmetro organizacional, o trabalho avalia o impacto destas mudanças em ambientes com características distintas.

Além disso, dada a falta de referências na área, também é objetivo deste trabalho propor uma metodologia preliminar para avaliação organizacional de SMA.

1.3 Metodologia

Nesta seção é apresentada a metodologia utilizada para condução desse trabalho, a qual está dividida em cinco etapas.

ETAPA 1 – SISTEMAS MULTIAGENTES CENTRADOS NA ORGANIZAÇÃO

Antes de avaliar como a organização dos agentes pode afetar o desempenho do sistema, é preciso entender qual a vantagem de se incorporar conceitos organizacionais, como grupos e papéis, em SMA. Além disso, o que muda em relação à programação do sistema com a adição de tais conceitos?

Desta forma, primeiramente foi realizada uma revisão bibliográfica sobre SMA centrados na organização, a fim de identificar a importância da inclusão de aspectos organizacionais na concepção de SMA.

ETAPA 2 – JACAMO E MAPC

Uma vez adotado o paradigma de SMA centrados na organização, deve-se escolher quais linguagens ou arcabouços devem ser utilizados na programação deste sistema. Assim, nesta etapa realizou-se um estudo aprofundado das funcionalidades e da arquitetura do arcabouço *JaCaMo*, com ênfase nos artefatos organizacionais que o compõem. Através desse estudo foi possível conhecer os diferentes parâmetros envolvidos na especificação de uma organização (papéis, grupos e missões), e como estes podem ser alterados.

Parte desse estudo consistiu na criação de um time para participação no *Multi-Agent Programming Contest 2012* (FRANCO; ROSSET; SICHMAN, 2013), o qual se mostrou um cenário interessante para a avaliação de diferentes organizações.

ETAPA 3 – AVALIAÇÃO ORGANIZACIONAL

Em seguida, foi feito um estudo bibliográfico sobre avaliação organizacional em SMA e métodos estatísticos empregados, tais como o teste de hipótese de “Wilcoxon Pareado”, utilizado neste trabalho.

ETAPA 4 – DEFINIÇÃO DOS EXPERIMENTOS

Nesta etapa foram definidos os experimentos, métricas e parâmetros organizacionais a serem utilizados. Variando-se então tais parâmetros, os diferentes times analisados neste trabalho puderam ser programados, e as métricas associadas colhidas durante a sua execução. Além disso, criou-se uma arquitetura para a execução dos experimentos e para a coleta dos dados de desempenho.

ETAPA 5 – ANÁLISE DOS RESULTADOS

Após a realização dos experimentos, os dados obtidos foram analisados utilizando testes estatísticos apropriados, cuja análise dos resultados é apresentada neste trabalho.

1.4 Organização do Texto

O restante deste documento está estruturado em cinco capítulos adicionais e um apêndice. O capítulo 2 apresenta os conceitos básicos para a construção de SMA, discute sobre a importância da organização em SMA, e discorre sobre o desenvolvimento de SMA com foco na utilização do arcabouço *JaCaMo*. O capítulo 3 introduz o *Multi-Agent Programming Contest*, e apresenta a arquitetura e estratégias utilizadas pelo time desenvolvido para participação nesta competição. Em seguida, no capítulo 4 mostra a importância da avaliação organizacional, como esta normalmente é realizada, e discorre sobre os métodos estatísticos utilizados. Os experimentos e resultados obtidos são apresentados no capítulo 5 e, por fim, as conclusões e trabalhos futuros estão presentes no capítulo 6.

No apêndice A são apresentados os resultados obtidos para os diferentes cenários testados neste trabalho.

2 SISTEMAS MULTIAGENTES

Nas últimas duas décadas, a proliferação de dispositivos de computação, bem como o advento da Internet, tem levado uma nova era de acessibilidade: pessoas, sistemas, e, sobretudo, a informação podem ser mais facilmente contatados e utilizados. Esta possibilidade de interconectar diferentes dispositivos e sistemas, e a necessidade em lidar com grandes massas de dados, tem feito crescer cada vez mais o interesse por sistemas abertos e distribuídos, em que seja possível reunir diferentes sistemas para a resolução de problemas de grande complexidade e/ou de natureza descentralizada.

Por outro lado, a dinâmica inerente aos sistemas abertos e a complexidade em gerir a interação entre um número elevado e heterogêneo de sistemas tem apontado a necessidade de introduzir algum grau de autonomia, de forma a permitir que os componentes respondam dinamicamente às evoluções no ambiente.

Neste contexto, SMA têm sido vistos como solução viável para o desenvolvimento de sistemas abertos e distribuídos, uma vez que fornecem técnicas e ferramentas para lidar com questões como a descentralização e distribuição do controle, interação e coordenação entre componentes, flexibilidade, segurança e abertura do sistema.

Desta forma, neste capítulo são introduzidos os conceitos básicos em SMA, e analisada a importância da organização em tais sistemas. Além disso, é discutido a programação de SMA, com foco para a arquitetura do arcabouço *JaCaMo*.

2.1 Conceitos Básicos

Os SMA são compostos por um conjunto de agentes autônomos capazes de interagir uns com os outros e com o ambiente, a fim de atingirem seus objetivos (WOOLDRIDGE, 2002). A ideia principal em SMA é que através da cooperação entre os agentes, é possível resolver problemas que estão além das capacidades individuais de

cada um.

Segundo o modelo AEIO (*Agent, Environment, Interaction, Organization*) (DEMAZEAU, 1995) um SMA é composto por 4 componentes básicos: agentes, ambiente, interação e organização. A Figura 1 ilustra a estrutura típica de um SMA, formada por diversos agentes que interagem entre si e são capazes de perceber e agir sobre diferentes partes do ambiente, de acordo com sua esfera de influência. Além disso, os agentes podem estar ligados/organizados através de diferentes relações, como as relações de poder, onde um agente pode ter autoridade sobre outros agentes (WOOLDRIDGE, 2002).

Figura 1: Estrutura de um Sistema Multiagente (JENNINGS; WOOLDRIDGE, 2001).

Nas seções a seguir são discutidos cada um desses componentes básicos para a criação de SMA.

2.1.1 Agentes

Segundo (WOOLDRIDGE, 2002), um *agente* é uma entidade computacional situada em algum ambiente sobre o qual é capaz de realizar ações de forma autônoma, de modo a alcançar os seus objetivos. A *autonomia* dos agentes significa que estes são capazes de agir sem a intervenção de seres humanos ou de outros sistemas, possuindo controle sobre o seu comportamento e estado interno (WEISS, 1999).

A Figura 2 mostra um agente e seu ambiente. Um agente consiste basicamente de um certo número de sensores e atuadores, além de um mecanismo de raciocínio e decisão. Os sensores são capazes de perceber o ambiente, criando percepções que

serão utilizadas pelo agente no raciocínio/decisão de qual ação executar, de maneira a alcançar seus objetivos. As ações dos agentes sobre o ambiente são efetuadas através dos seus atuadores (RUSSELL; NORVIG, 2010).

Figura 2: Agentes interagem com o ambiente através de sensores e atuadores (RUSSELL; NORVIG, 2010).

O comportamento apresentado pelos agentes, a fim de alcançar seus objetivos, é resultante de suas arquiteturas de funcionamento interno. Basicamente, existem dois tipos de arquiteturas: agentes *reativos* e agentes *cognitivos* (HÜBNER, 2003).

Os agentes reativos escolhem suas ações com base unicamente nas percepções correntes que possuem do ambiente, sem raciocinar sobre elas. Assim, o comportamento do agente é definido unicamente através de regras do tipo: condição $\rightarrow$ ação, que simplesmente mapeiam as percepções do ambiente em ações (WOOLDRIDGE, 2002). Uma arquitetura possível, entre outras, para agentes reativos é descrita na Figura 3.

Figura 3: Agente reativo (RUSSELL; NORVIG, 2010).

Considerando este funcionamento bastante simples, este tipo de agente apresenta em geral as seguintes propriedades: normalmente possui representação do conhecimento implícita no código; por não possuir memória, não tem histórico dos fatos que aconteceram e das ações que executou; e não tem controle deliberativo (planejamento) de suas ações.

Por sua vez, os agentes cognitivos possuem um estado mental e funcionam racionalmente, isto é, raciocinam para construir um plano de ações que leva a um objetivo pretendido (Figura 4). Portanto, apresentam características particulares que os diferenciam de programas convencionais e dos agentes reativos. Dentre elas, os agentes: possuem autonomia para alterar seu funcionamento a fim de adaptar-se melhor ao ambiente; estão continuamente em funcionamento; possuem representação do conhecimento explícita no código (conhecimento introspectivo); o mecanismo de controle é deliberativo (o agente raciocina sobre que ações realizar); têm memória e portanto podem lembrar de ações realizadas no passado.

Figura 4: Agente cognitivo (DEMAZEAU; MÜLLER, 1990).

Um modelo de arquitetura amplamente utilizado para agentes cognitivos é a BDI (*Beliefs-Desires-Intentions*) (BRATMAN, 1987), que tem suas raízes no estudo do processo humano de decidir, momento a momento, que ação executar de forma a alcançar os seus objetivos. Esse processo é chamado de raciocínio prático (*practical reasoning*).

O raciocínio prático envolve dois importantes processos: decidir que objetivos se deseja alcançar, e como se irá alcançá-los. O primeiro processo é chamado de deliberação, e o segundo de raciocínio “meio-fim” (ou *means-ends*) (WEISS, 1999).

A arquitetura BDI é formada por três estruturas básicas que representam as crenças, desejos e intenções do agente, e funções responsáveis pela deliberação e pelo raciocínio meio-fim. A Figura 5 apresenta um diagrama simples para a arquitetura BDI. Durante um ciclo percepção-ação, o agente atualiza a sua base de crenças com as percepções do ambiente, e escolhe entre seus desejos quais objetivos deverá tentar alcançar; tais objetivos são denominados de intenções. Uma vez definidas as intenções, o agente então raciocina sobre qual ação deverá executar. Esse raciocínio sobre como alcançar o objetivo, em geral, resulta em um plano a ser seguido pelo agente.

Figura 5: Agente BDI.

Uma propriedade importante em agentes BDI é que as intenções são persistentes, ou seja, o agente se compromete com uma intenção até que ela seja realizada, ou até que fique claro que ela nunca poderá ser alcançada. Caso não houvesse esta propriedade de persistência, existiria a possibilidade de um objetivo nunca ser concluído, uma vez que outras intenções tomariam o seu lugar em cada ciclo de raciocínio do agente.

Neste trabalho, adotou-se uma arquitetura BDI para os agentes através da utilização do arcabouço *JaCaMo*, o qual é detalhado na seção 2.3.

2.1.2 Ambientes

O ambiente em que deverá atuar afeta diretamente o projeto de um agente. Russel e Norvig sugerem a seguinte classificação para as propriedades do ambiente que podem interferir na concepção dos agentes e, consequentemente, do SMA (RUSSELL; NORVIG, 2010):

- **Totalmente observável vs. Parcialmente observável:** Se um agente consegue, através dos seus sensores, ter acesso a uma visão completa do ambiente em qualquer instante no tempo, então dizemos que o ambiente é totalmente observável, e neste caso, o agente não precisa manter internamente nenhum tipo de representação do mundo. Um ambiente pode ser parcialmente observável devido ao ruído, aos sensores imprecisos, ou por que parte do ambiente não pode ser observado diretamente através dos seus sensores.

Quanto mais observável um ambiente, mais simples é construir agentes que operem eficientemente nele. Um “bom” agente é aquele que toma as decisões certas, e estas decisões são claramente dependentes da qualidade de informações disponíveis ao agente.

- **Determinístico vs. Estocástico:** Se não existe incerteza sobre o estado que irá resultar da realização de uma ação, dizemos que o ambiente é determinístico, e estocástico (ou não-determinísticos) em caso contrário.

Do ponto de vista do projeto de agentes, ambientes determinísticos são preferíveis a ambientes estocásticos. Se não há qualquer incerteza sobre o resultado de uma ação, o agente não precisa verificar qual foi o seu resultado, ou reconsiderar o plano em curso. Em particular, em um ambiente determinístico, o projeto do agente pode assumir que as ações executadas pelo agente sempre terão êxito.

- **Episódico vs. Sequencial:** Em um ambiente episódico, a experiência do agente é dividida em episódios atômicos. Em cada episódio, o agente recebe as percepções do ambiente e então executa uma só ação. Além disso, o próximo episódio não depende das ações efetuadas em episódios anteriores. Em ambientes sequenciais, por outro lado, as decisões anteriores do agente afetam futuras decisões, ou seja, ações de curto prazo podem ter consequências a longo prazo.

Ambientes episódicos são mais simples de se lidar do que ambientes sequenciais, pois nos primeiros o agente não precisa raciocinar sobre o futuro.

- **Estático vs. Dinâmico:** Um ambiente estático é aquele que permanece inalterado, exceto pela realização de uma ação pelo agente, enquanto que ambientes dinâmicos são aqueles em que o estado pode mudar independentemente da vontade do agente.

Em SMA, a presença de vários agentes faz com que o ambiente seja dinâmico do ponto de vista de cada agente. Assim, é importante que os agentes sincronizem e coordenem suas ações com os outros agentes, a fim de alcançarem os seus objetivos.

- **Discreto vs. Contínuo:** Um ambiente discreto é aquele em que existe um número finito e fixo de estados distintos, em contraste com os ambientes contínuos, onde o número de estados pode ser incontável.

Projetar agentes para ambientes discretos é mais simples, pois a princípio é possível enumerar todos os possíveis estados do ambiente e determinar a ação ótima a ser executada em cada um desses estados. No entanto, tal abordagem é raramente utilizada, sendo até mesmo inviável em ambientes com um número finito mas muito grande de estados.

Neste trabalho, o ambiente do MAPC pode ser considerando como sendo: parcialmente observável, estocástico, episódico, dinâmico e contínuo. Mais detalhes sobre o ambiente podem ser encontrados no capítulo 3.

2.1.3 Interações

Em SMA, agentes devem ser capazes de interagir uns com os outros, de forma a satisfazer os seus objetivos. Assim, é essencial que exista uma infraestrutura que permita que tais interações entre os agentes aconteçam.

Essa infraestrutura deve incluir protocolos para a comunicação e para a interação dos agentes. Protocolos de comunicação permitem que os agentes troquem e entendam mensagens, enquanto, protocolos de interação permitem que os agentes mantenham uma conversa (WEISS, 1999).

Um aspecto importante em SMA que decorre da interação entre os agentes, seja em situações de cooperação ou de competição, é a coordenação de tarefas e objetivos. No caso de cooperação, vários agentes tentam combinar os seus esforços para realizar, como um grupo, aquilo que não é possível individualmente. Já em ambientes competitivos, os agentes trabalham uns contra os outros, procurando maximizar os seus próprios ganhos à custa dos outros agentes. Neste caso, o sucesso de um agente representa o não-sucesso dos outros. Nas duas situações, seja de cooperação ou de

competição, os agentes devem levar em consideração as dependências existentes entre suas atividades para a coordenação dos seus objetivos e tarefas.

Neste trabalho, a interação entre os agentes é realizada através do arcabouço *Ja-CaMo*, o qual é apresentado na seção 2.3.

2.1.4 Organizações

A organização fornece uma estrutura para a interação dos agentes através da definição de papéis, expectativas de comportamento e de relações de autoridade (SYCARA, 1998). Ela restringe o comportamento dos agentes, associando-os a papéis que indicam quais tarefas cada um deve realizar, e impondo padrões de cooperação de forma a guiar o SMA a alcançar o comportamento global desejado.

Neste trabalho, o foco principal é estudar como a organização do SMA pode afetar o seu desempenho. A seguir, descrevem-se em mais detalhes as organizações em SMA.

2.2 Organizações em Sistemas Multiagentes

Uma questão importante em SMA é como garantir alguma ordem no sistema de tal forma que a autonomia dos agentes não leve a um comportamento coletivo indesejado ou caótico, retardando ou impedindo que os objetivos globais sejam alcançados.

Para lidar com esse problema, é necessário algum meio de estruturação da sociedade, de modo a reduzir a complexidade do sistema, aumentar sua eficiência, e de forma mais precisa modelar o problema em questão (JENNINGS; WOOLDRIDGE, 2001). Assim, a organização em SMA é um aspecto fundamental para restringir o comportamento dos agentes e obter um funcionamento coerente do sistema, permitindo que os objetivos globais sejam alcançados mais facilmente (HÜBNER; SICHMAN; BOISSIER, 2002; HÜBNER; SICHMAN; BOISSIER, 2006).

2.2.1 Origem e Visibilidade das Organizações

Os modelos usados em SMA para descrever ou projetar uma organização são classicamente divididos em dois pontos de vista: centrada no agente e centrada na orga-

nização (LEMAÎTRE; EXCELENTE, 1998). Na primeira abordagem, não existe uma representação explícita da organização. Esta emerge como resultado da interação entre os agentes. Já no segundo ponto de vista, a organização existe a priori (definida pelo projetista do sistema ou pelos próprios agentes) e deve ser seguida pelos agentes (SICHMAN; DIGNUM; CASTELFRANCHI, 2005).

Os SMA desenvolvidos com visão no agente são chamados de ACMA (Agent-Centered Multi-Agent Systems), enquanto aqueles com especificação explícita da organização são chamados de OCMA (Organization-Centered Multi-Agent Systems).

Nos ACMA, o foco do desenvolvedor está em definir as ações que os agentes poderão desempenhar, e como estes irão escolhê-las, face às mudanças no ambiente e às suas interações com os outros agentes. Assim, o SMA é todo definido com base nos estados mentais de seus agentes (objetivos, crenças, planos) e da relação entre esses estados e o comportamento global do sistema.

Por outro lado, nos OCMA o foco maior encontra-se na organização, isto é, na estrutura do SMA. O sistema é definido não mais a partir dos estados mentais dos agentes, mas através de conceitos organizacionais, tais como papéis, grupos e tarefas (FERBER; GUTKNECHT; MICHEL, 2004). Assim, é possível definir a estrutura do sistema sem especificar detalhes sobre a implementação dos agentes, o que é particularmente interessante em SMA abertos, onde não se conhece a priori a arquitetura dos agentes.

Além disso, os agentes podem ter conhecimento ou não da organização do SMA. Em alguns casos, os agentes não sabem que fazem parte de uma organização, pois ou ela emergiu das suas interações ou foi codificada diretamente, de maneira que estes não são capazes de raciocinar sobre ela. Em outros casos, os agentes tem conhecimento e podem raciocinar sobre a organização: em certos casos, ela foi explicitamente definida e está acessível aos agentes e, em outros, os agentes são capazes, com o tempo, de criar mentalmente uma representação interna da organização, através de suas percepções do ambiente e das suas interações com os outros agentes (BOISSIER; HÜBNER; SICHMAN, 2007).

Este trabalho de avaliação de desempenho organizacional é feito sobre SMA com organizações explícitas e acessíveis aos agentes, pois, como veremos na seção a seguir, estes trazem algumas vantagens em relação aos SMA tradicionais com visão centrada no agente.

2.2.2 SMA Centrados nas Organizações

A organização em OCMAS pode ser vista como um conjunto de restrições que um grupo de agentes adota, de modo a controlar a autonomia dos agentes, e fazendo com que o sistema alcance mais facilmente os seus objetivos globais.

Além disso, como apresentado em (FERBER; GUTKNECHT; MICHEL, 2004), a falta de uma organização explícita faz com que os ACMAS apresentem alguns inconvenientes, tais como:

- Prever o comportamento de todo o sistema com base nos componentes constituintes (os agentes) é extremamente difícil (por vezes impossível). A falta de uma descrição da organização dificulta o raciocínio sobre o funcionamento do SMA.
- A possibilidade que todos os agentes possam interagir, sem qualquer controle externo, pode acarretar problemas de segurança. É responsabilidade de cada agente verificar a qualificação de seus interlocutores e restringir seu acesso a partir de outros agentes, mas não há garantia de que um agente irá realmente considerar tais restrições. Desta forma, é fácil para um agente malévolo agir incorretamente e usar o sistema de forma fraudulenta.

Desta forma, os OCMAS trazem diversas vantagens em relação aos ACMAS, principalmente no que diz respeito à criação de SMA abertos mais seguros e eficientes.

2.2.3 Paradigmas Organizacionais

Conhecer o espectro de possíveis paradigmas organizacionais, suas diferenças, seus pontos fortes e fracos, é muito importante para seleção da organização adequada para um determinado domínio e ambiente. *Computation Organization Theory* (COT) (CARLEY, 1995; PRIETULA; CARLEY; GASSER, 1998; CARLEY; GASSER, 1999) é uma área de pesquisa interdisciplinar cujos membros se concentram em desenvolver e analisar modelos organizacionais, com foco na busca dos princípios gerais de organização e das condições em que estes princípios se aplicam ou não.

Diferentes trabalhos nessa área (HORLING; LESSER, 2004; DIGNUM, 2004; ARGENTE; JULIAN; BOTTI, 2006) têm analisado e classificado as organizações.

Segundo a classificação de Horling (HORLING; LESSER, 2004) os paradigmas organizacionais mais comumente empregadas em SMA são: hierarquia, coalizão, time e mercado.

Em uma **hierarquia**, os agentes cooperam a fim de alcançar um objetivo comum e são dispostos em árvore, de modo que a interação entre os agentes é limitada as relações diretas entre superior e subordinados, como na Figura 6.

Figura 6: Exemplo de hierarquia. (HORLING; LESSER, 2004)

Agentes no topo da árvore comandam os agentes localizados abaixo, que são benevolentes: sempre realizam as tarefas que lhes foram incumbidas. Os dados produzidos pelos agentes nos níveis mais baixos viajam para os níveis superiores na hierarquia, a fim de fornecer uma visão mais ampla, enquanto o controle flui dos níveis superiores para os inferiores.

Hierarquias são adequadas em ambientes onde o objetivo é a produção eficiente de algum resultado ou produto. Esses ambientes exigem um controle confiável de recursos e do fluxo de informação através de entidades centrais (DIGNUM, 2004). Hierarquias só são úteis em ambientes onde é possível a decomposição de tarefas. Assim, ao receber uma nova tarefa o agente, caso não seja capaz de realizá-la por completo, pode procurar um ou mais agentes capazes de ajudá-lo e, desta forma, decompor a tarefa em sub-tarefas.

Já as **coalizões** diferem de outros paradigmas organizacionais por serem apenas necessárias enquanto há um objetivo em comum a ser realizado. Coalizões são formadas tendo como base fundamental um objetivo comum entre os participantes da coalizão em questão e são dissolvidas quando o objetivo já não mais existe ou é impossível de ser realizado. Além disso, a estrutura da coalizão é plana, ou seja, não há uma hierarquia definida dentro da coalizão. Não existem líderes ou agentes que possam decidir

as ações dos demais quando há um impasse.

Figura 7: Exemplo de coalizões. (HORLING; LESSER, 2004)

Agentes em uma coalizão podem possuir objetivos próprios, mas estão dispostos a trocar um pouco da sua liberdade para formar relações de confiança com outros agentes. Desta forma, coalizões são adequadas em ambientes onde é necessário existir colaboração entre os agentes.

Mercados são organizações cujos agentes possuem interesses próprios e negociam com os outros agentes no sentido de satisfazer seus próprios objetivos. Neste tipo de organização existem dois tipos de papéis: os compradores e os vendedores, como apresentado na Figura 8. Os compradores podem requerer ou dar lances em um conjunto de itens que os interessam, tais como recursos ou serviços. Os vendedores recebem e processam as propostas determinando os vencedores. Este arranjo cria um sistema do tipo produtor/consumidor como o encontrado nos mercados econômicos do mundo real.

Figura 8: Exemplo de mercado. Os compradores são representados em branco e os vendedores em preto. (HORLING; LESSER, 2004)

Os mercados são particularmente adequados em situações em que os recursos se sobrepõem e os agentes precisam competir por eles, sendo assim uma boa opção para problemas de alocação de recursos ou serviços (DIGNUM, 2004).

Por fim, um **time** consiste de um grupo de agentes que trabalham em conjunto

para alcançar um objetivo em comum. Em geral, cada agente assume um papel e fica responsável por realizar algumas das sub-tarefas necessárias para se atingir o objetivo maior do time. Esses papéis podem mudar ao longo do tempo em resposta a eventos e mudanças no ambiente, enquanto o próprio objetivo de alto nível se mantém constante.

Figura 9: Exemplo de time. (HORLING; LESSER, 2004)

Times normalmente mantêm uma representação explícita dos seus objetivos, crenças comuns e dos seus planos globais. Este tipo de representação oferece flexibilidade e robustez, permitindo que os agentes raciocinem explicitamente sobre os seus comportamentos num nível coletivo.

O principal benefício de um time é que, agindo em conjunto, o grupo de agentes pode enfrentar problemas maiores que não poderiam ser realizados por um agente apenas. Assim, os times são especialmente apropriados para a realização de tarefas complexas com muitas sub-tarefas interdependentes.

Neste trabalho, foi utilizado o paradigma organizacional de time, visto que este é o mais adequado dentro do contexto do *Multi-Agent Programming Contest*, onde diferentes agentes devem cooperar durante toda a competição a fim de dominar as maiores áreas do mapa, como apresentado no capítulo 3.

2.3 Programação de Sistemas Multiagentes

Como visto na seção 2.1, a abordagem AEIO define um SMA como sendo formado por 4 componentes básicos: agentes, ambiente, interação e organização. Deste modo, o desenvolvedor ao projetar um SMA deve, em resumo, se preocupar em definir (não necessariamente nesta ordem): (i) a arquitetura e comportamento dos agentes; (ii) as linguagens e protocolos de interação; (iii) a arquitetura do ambiente; e (iv) a organização do SMA. Assim, nesta seção será discutido o desenvolvimento de cada um desses componentes básicos para a construção de SMA.

2.3.1 Programação de Agentes

Suponha que desejemos implementar um agente baseado no modelo BDI (descrito anteriormente na seção 2.1), fazendo uso de deliberação e raciocínio meio-fim. Como deveria ser programado um tal agente? Em uma primeira tentativa, poderíamos imaginar um laço de controle representado no Algoritmo 2.1, onde o agente continuamente (i) observa o mundo (o ambiente) e atualiza sua base de crenças; (ii) delibera para decidir qual intenção deve alcançar; (iii) usa raciocínio meio-fim para encontrar um plano para atingir a intenção; e (iv) executa o plano (BORDINI; HÜBNER; WOOLDRIDGE, 2007).

Algoritmo 2.1 Laço de controle simples para um agente BDI.

```

1:  $B \leftarrow B0$  ▷ crenças iniciais
2: while true do
3:   obtêm a próxima percepção  $\rho$  através dos sensores;
4:    $B \leftarrow brf(B, \rho)$ ; ▷ atualização da base de crenças
5:    $I \leftarrow delibera(B)$ ;
6:    $\pi \leftarrow planeja(B, I)$ ;
7:   executa( $\pi$ );
8: end while

```

No entanto, este laço de controle simples apresenta alguns problemas. A questão principal tem a ver em como o agente se compromete com uma intenção (ou plano). Neste algoritmo básico, o agente é fortemente comprometido, isto é, uma vez escolhida uma intenção, o agente continua comprometido a ela até que tenha executado completamente o plano para alcançá-la, mesmo que durante a execução do plano esta intenção tenha se tornado inalcançável. Da mesma forma, uma vez selecionado um plano, o agente permanece comprometido a ele, mesmo que o estado do ambiente tenha mudado e tornado o plano inútil.

Com este problema em mente podemos, após alguma reflexão, chegar à versão refinada do laço de controle do agente, apresentada no Algoritmo 2.2. Com este novo laço de controle, o agente permanece comprometido ao seu plano até que: (i) ele tenha sido totalmente executado; (ii) a intenção tenha sido alcançada ou se acredita ser inatingível; ou (ii) acredita-se que o plano não tem mais qualquer utilização.

As variáveis B , D e I representam, respectivamente, as crenças, desejos e intenções do agente. Assim como no laço simples, o agente observa o ambiente para obter as novas percepções, para em seguida atualizar a sua base de crenças (linha 5) através

Algoritmo 2.2 Laço de controle mais refinado para um agente BDI (BORDINI; HÜBNER; WOOLDRIDGE, 2007).

```

1:  $B \leftarrow B0$                                 ▶ crenças iniciais
2:  $I \leftarrow I0$                                 ▶ intenções iniciais
3: while true do
4:   obtêm a próxima percepção  $\rho$  através dos sensores;
5:    $B \leftarrow bnf(B, \rho);$                         ▶ atualização da base de crenças
6:    $D \leftarrow opcoes(B, I);$ 
7:    $I \leftarrow filtra(B, D, I);$ 
8:    $\pi \leftarrow plano(B, I, Ac);$                     ▶ Ac é a lista de ações
9:   while not (vazio( $\pi$ ) ou alcançado( $I, B$ ) ou impossível( $I, B$ )) do
10:     $\alpha \leftarrow$  primeiro elemento de  $\pi$ ;
11:    execute( $\alpha$ );
12:     $\pi \leftarrow \pi - \alpha$ ;
13:    observa o ambiente para obter a próxima percepção  $\rho$ ;
14:     $B \leftarrow bnf(B, \rho);$ 
15:    if reconsiderar( $I, B$ ) then
16:       $D \leftarrow opcoes(B, I);$ 
17:       $I \leftarrow filtra(B, D, I);$ 
18:    end if
19:    if invalido( $\pi, I, B$ ) then
20:       $\pi \leftarrow plano(B, I, Ac);$ 
21:    end if
22:  end while
23: end while

```

da função *brf* (*belief revision function*).

Na linha 6, o agente determina quais são seus desejos, ou opções, com base nas suas crenças e intenções atuais. O agente então seleciona algumas opções para se transformarem em intenções (linha 7), e assim gerar um plano para alcançá-las (linha 8).

O laço na linha 9 corresponde à execução do plano. Se tudo estiver correndo bem, o agente simplesmente remove a cada interação uma ação do plano e a executa, até que o plano fique vazio, ou seja, até que todas as ações tenham sido executadas. No entanto, após executar uma ação (linha 11), o agente volta a observar o ambiente e atualiza a sua base de crenças. Ele então verifica se deve ou não reconsiderar suas atuais intenções (linha 15), dadas as mudanças observadas no ambiente.

Por fim, o agente se pergunta se o plano atual continua válido com relação as suas intenções e crenças (linha 19). Caso considere que aquele plano não é mais válido, ele replaneja (linha 20).

Existem diferentes plataformas para o desenvolvimento de SMA que implementam a arquitetura BDI representada no Algoritmo 2.2, tais como JACK (HOWDEN et al., 2001), Jadex (POKAHR; BRAUBACH; LAMERSDORF, 2005), Jason (BORDINI; HÜBNER; WOOLDRIDGE, 2007) e SPARK (MORLEY; MYERS, 2004). Desta forma, ao programar os agentes o desenvolvedor pode fazer uso direto da arquitetura fornecida, tendo apenas que se preocupar em definir as crenças, metas e intenções iniciais do agente, as ações possíveis sobre o ambiente, e os planos para a realização de suas intenções. O desenvolvedor, portanto, trabalha em determinar o comportamento dos agentes, enquanto os arcabouços fornecem a arquitetura necessária para a sua execução.

Um exemplo de arcabouço para o desenvolvimento de SMA, denominado *Jason*, é apresentado a seguir. Tal arcabouço foi utilizado neste trabalho.

2.3.1.1 Jason

Jason é uma plataforma para o desenvolvimento de SMA que interpreta uma versão estendida da linguagem de comunicação de agentes *AgentSpeak* (RAO, 1996), permitindo a implementação dos agentes através da arquitetura BDI.

Um agente é definido em *Jason* através de:

- (i) um conjunto inicial de crenças (*beliefs*) que representam o conhecimento inicial do agente sobre o mundo,
- (ii) um conjunto de metas (*desires*),
- (iii) um conjunto de planos que o agente pode instanciar e executar dinamicamente para alcançar seus objetivos (*intentions*).

Os planos são descritos por regras do tipo: ***Evento : Contexto $\leftarrow$ Corpo*** (como em Prolog), onde ***Evento*** representa o evento de desencadeamento do plano, tais como a adição de uma nova crença (+b), um objetivo (+!g), ou a falha na execução de um plano (-!g). O ***Contexto*** consiste de uma expressão lógica a ser aplicada sobre a base de crença do agente, e que define as condições em que o plano pode ser executado. O ***Corpo*** do plano contém a sequência de ações a serem executadas, tais como: criação de submetas (!g), atualização da base de crenças (+b, -b), e ações externas a serem executadas sobre o ambiente.

Tabela 1: Tipos de Eventos

Notação	Tipo
+b	adição de uma crença
-b	remoção de uma crença
+!g	adição de um objetivo
-!g	remoção de um objetivo

Desta forma, ao ocorrer um evento, o contexto é comparado com o estado atual do agente e, caso as condições sejam satisfeitas, o corpo do plano é executado. Caso contrário, continua-se a busca por um plano que possa ser executado para aquele evento; se nenhum plano for aplicável, o evento falha. Um exemplo de planos em *AgentSpeak* é apresentado no Código 2.1.

Código 2.1: Exemplo de planos em *AgentSpeak* (BORDINI; HÜBNER, 2007).

```
// adição de um novo concerto na base de crenças do agente
// A = artista, L = local
+concerto(A,L) : gosta(A)
  <- !comprar_ingressos(A,L).    // nova meta (objetivo)

// plano para compra de ingressos
+!comprar_ingressos(A,L) : not ocupado(telefone)
  <- telefonar(L);              // ação
  ...;
  !escolher_assento(A,L).        //submeta
```

2.3.2 Programação de Ambientes

O ambiente é um conceito básico em SMA, e representa o lugar físico ou topológico onde os agentes estão situados. Ele corresponde ao mundo externo percebido pelos agentes, e onde os agentes executam ações, de modo a cumprir seus objetivos.

Desta forma, parte do desenvolvimento de um SMA está em projetar como o ambiente irá se comportar e como os agentes irão interagir com ele. O desenvolvedor deve assim definir como o agente irá perceber o ambiente, o que ele será capaz de perceber, e como as suas ações irão afetar o ambiente. Além disso, o ambiente também pode encapsular funcionalidades e serviços para apoiar as atividades dos agentes.

Este último ponto é explorado pelo arcabouço *CartAgO*, o qual foi utilizado neste trabalho e é apresentado a seguir.

2.3.2.1 CArtAgO

CartAgO (RICCI; VIROLI; OMICINI, 2007) (*Common Artifact Infrastructure for Agent Open environment*) é um arcabouço baseado no meta-modelo A&A (Agentes e Artefatos) (OMICINI; RICCI; VIROLI, 2008) para o desenvolvimento e execução de ambientes em SMA. A ideia principal é que o ambiente deve ser visto como uma camada computacional capaz de encapsular funcionalidades e serviços que os agentes podem explorar em tempo de execução (BOISSIER et al., 2011).

No *CartAgO*, o ambiente é projetado como um conjunto dinâmico de entidades computacionais chamadas de *artefatos*, que representam os recursos e ferramentas que os agentes são capazes de criar, descobrir, perceber, e usar.

O conjunto de artefatos é organizado em um ou múltiplos *workspaces* (áreas de trabalho), que podem ser distribuídos em vários nós de uma rede. Agentes podem entrar e sair dos *workspaces* e, dentro destes, artefatos podem ser criados e descartados dinamicamente pelos agentes (RICCI; PIUNTI; VIROLI, 2010). Os *workspaces* são usados para representar a topologia do sistema a partir de um ponto de vista lógico. Na Figura 10, dois *workspaces* são representados e mapeados em três nós de uma rede. Agentes e artefatos podem pertencer a vários *workspaces* e, além disso, o mesmo artefato pode ser distribuído em vários nós.

Cada artefato possui um manual, utilizado para descrever as suas funcionalidades,

Figura 10: Representação dos *workspaces* (RICCI; VIROLI; OMICINI, 2006).

além de um conjunto de operações e propriedades observáveis a serem utilizados pelos agentes. A execução de uma operação pode gerar um sinal, ou a atualização de uma propriedade observável. Além disso, artefatos podem ser conectados entre si, de modo a permitir que um artefato dispare a realização de uma operação em outro artefato.

Figura 11: Representação de um artefato (RICCI; PIUNTI; VIROLI, 2010).

Em resumo, o *CARtAgO* fornece (i) a capacidade de definir e programar novos tipos de artefatos; (ii) uma API para os agentes trabalharem com artefatos e *workspaces*; e (iii) uma plataforma de execução e gerenciamento de ambientes baseados em artefatos (HÜBNER et al., 2009; KITIO et al., 2008).

2.3.3 Programação de Interações

Parte importante no desenvolvimento de SMA consiste em definir como os agentes irão interagir e trocar informações. Para isso, devem-se utilizar protocolos de comunicação e interação que permitam que os agentes troquem, entendam mensagens e consigam manter um diálogo uns com os outros.

Ao projetar a comunicação entre os agentes, deve-se dar atenção a diferentes níveis de abstração do sistema. Por exemplo, em um nível mais baixo, de rede, deve-se ter certeza que as mensagens transmitidas entre os agentes chegam de forma segura e rapidamente ao seu destino. Em um nível intermediário, de linguagem, deve-se definir um conjunto básico de primitivas a serem trocadas entre os agentes, de forma que eles possam facilmente se entender. Por fim, em um nível mais alto, de aplicação, deve-se fazer uso da comunicação como meio de se atingir coordenação entre os agentes (NIKOS, 2007).

Os arcabouços para o desenvolvimento de SMA, normalmente, fornecem a infraestrutura necessária para que ocorra interação entre os agentes, facilitando assim o trabalho do desenvolvedor, que pode então focar em como utilizar efetivamente essa comunicação para a coordenação dos agentes.

2.3.3.1 Comunicação em Jason

No arcabouço *Jason*, os agentes são capazes de se comunicar usando a linguagem de comunicação de agentes KQML (FININ et al., 1994). Ela permite que um agente peça ajuda a outros agentes para alcançar um objetivo (*achieve*); informe outros agentes sobre mudanças no ambiente (*tell*); e até mesmo, peça informação sobre a base de crenças de um outro agente (*askOne*) (JENSEN, 2010).

As mensagens são implementadas como ações a serem realizadas pelos agentes e funcionam em conjunto com a linguagem *AgentSpeak*, tornando a comunicação muito fácil de usar. Assim, pode-se retomar o exemplo anterior, apresentado na seção 2.3.1.1, adicionando a ele as ações em *AgentSpeak* responsáveis pela interação entre os agentes, como no Código 2.2.

Código 2.2: Exemplo de comunicação em *AgentSpeak*.

```

// adição de um novo concerto na base de crenças do agente
// A = artista, L = local
+concerto(A,L) : gosta(A)
  <- !comprar_ingressos(A,L).    // nova meta (objetivo)

// plano para compra de ingressos
+!comprar_ingressos(A,L) : not ocupado(telefone)
  <- telefonar(L,N);           // ação
    .send(N,tell,bomDia(N));
    .send(N,askOne,preço(A,L,P),preço(A,L,P));
    ...;
    !escolher_assento(A,L).    //submeta

```

2.3.4 Programação de Organizações

Como visto na seção 2.2, organizações podem emergir da interação entre os agentes, ou podem ser explicitamente especificadas. Neste trabalho, o interesse encontra-se em SMA onde a organização é quem guia o desenvolvimento do sistema, além de ter sido explicitamente especificada. Desta forma, abordaremos nesta seção apenas a programação de OCMAS.

2.3.4.1 Programação de OCMAS

A programação de um OCMAS se inicia pela descrição explícita da organização. Esta descrição é feita através de uma *Linguagem de Modelagem de Organizações* (LMO), com a qual é possível especificar as restrições e padrões de cooperação impostas aos agentes pelo programador (ou pelos agentes), resultando em uma representação explícita chamada de *Especificação Organizacional* (EO). O conjunto de agentes alocados em uma instância da EO forma uma *Entidade Organizacional* (EnO) (BOISSIER; HÜBNER; SICHTMAN, 2007).

A EO é executada e interpretada através de uma *Infraestrutura Organizacional* (IO). A IO tem duas finalidades: (i) ajudar os agentes a realizar os objetivos globais do SMA, proporcionando-lhes o acesso aos serviços da organização; e (ii) controlar e fazer cumprir as restrições impostas na organização, de forma a garantir que os agentes se comportem de acordo com a EO (HÜBNER et al., 2009).

Vários são os exemplos de LMO obtidos na literatura, como ISLANDER (ESTEVA; CRUZ; SIERRA, 2002) e OperA (DIGNUM, 2004). A primeira contempla uma respectiva IO, denominada AMELI (ESTEVA et al., 2004), enquanto a se-

gunda fornece um simulador organizacional, chamado OperettA (OKOUYA; DIGNUM, 2008).

Este trabalho utilizará o modelo organizacional *Moise*, descrito a seguir.

2.3.4.2 Modelo Organizacional Moise

Moise (Model of Organization for multi-agent SystEms) é uma LMO que decompõe a especificação da organização em três dimensões: estrutural, funcional e deôntica (HÜBNER; SICHTMAN; BOISSIER, 2002).

A *Especificação Estrutural* (EE) define os papéis, relações entre papéis, e grupos de uma organização. Cada agente pode assumir um ou mais papéis, e cada papel está relacionado com um conjunto de restrições comportamentais que um agente aceita ao entrar em um grupo. As relações entre os papéis são definidas através de ligações, que podem ser de três tipos: (i) ligações de conhecimento, que determinam se agentes desempenhando um papel têm permissão de conhecerem agentes que desempenham outros papéis; (ii) ligações de comunicação, que definem com quais outros agentes um agente pode se comunicar, dados os papéis que desempenham; e (iii) ligações de autoridade, que estabelecem se um determinado papel dá ao agente autoridade ou não sobre outros agentes desempenhando outros papéis. Finalmente, ao nível de grupo é possível determinar: os papéis disponíveis no grupo e seus subgrupos, compatibilidade entre papéis, e número mínimo e máximo de agentes que podem adotar um determinado papel dentro do grupo.

A *Especificação Funcional* (EF) define como os objetivos globais devem ser alcançados, isto é, como esses objetivos são decompostos (em planos) e distribuídos aos agentes (em missões). A EF consiste de uma árvore de decomposição de metas globais, conhecida como *Esquema Social* (SCH), onde a raiz é a meta do esquema e os nós as sub-metas. Metas e sub-metas são divididas em missões a serem distribuídas aos agentes. O SCH também especifica como as sub-metas estão relacionadas, se devem ser alcançadas em paralelo, em sequência, ou se existe a possibilidade de escolha entre duas ou mais sub-metas.

A *Especificação Deôntica* (ED) liga as especificações estrutural e funcional definindo com quais missões um papel tem a obrigação ou a permissão de se comprometer.

O modelo *Moise* é, portanto, um modelo centrado na organização onde as duas

primeiras dimensões podem ser especificadas de forma quase independente e depois ligadas pela dimensão deôntica. Tal independência permite que o SMA altere a sua estrutura sem mudar o seu funcionamento, e vice-versa, sendo apenas necessário o ajuste das suas relações deônticas (HÜBNER; SICHMAN; BOISSIER, 2006).

Em relação a IO, existem alguns arcabouços que suportam o modelo *Moise*, tais como: S-Moise+ (HÜBNER; SICHMAN; BOISSIER, 2006), J-Moise+ (HÜBNER; SICHMAN; BOISSIER, 2007), e ORA4MAS (HÜBNER et al., 2009).

Por fim, existe uma representação gráfica para cada uma das especificações do *Moise*, cujos exemplos podem ser encontrados na seção 3.2.2.1.

Em resumo, este trabalho utilizou de forma combinada os arcabouços *Jason* e *CARtAgO*, descritos respectivamente nas seções 2.3.1.1 e 2.3.2.1, bem como a IO ORA4MAS. Para tal, fez-se uso da infraestrutura fornecida pelo arcabouço *JaCaMo*, que combina tais ferramentas, e que é descrito a seguir.

2.3.5 Arcabouço JaCaMo

Nem todos os arcabouços para o desenvolvimento de SMA fornecem ferramentas para o desenvolvimento dos componentes apresentados até aqui (agentes, ambiente, interação e organização). A sua maioria foca apenas em fornecer uma infraestrutura para a implementação e execução dos agentes, não tratando da programação da organização, ou mesmo do ambiente.

O *JaCaMo* (BOISSIER et al., 2011) é um arcabouço que cobre todos os níveis de abstração necessários para o desenvolvimento de sofisticados SMA, pois possibilita, além da programação dos agentes, a definição do ambiente e da organização do SMA. Cada um desses níveis de abstração (agente, ambiente e organização) são implementados através de um dos três componentes já discutidos nas seções anteriores que compõem o arcabouço: *Jason*, *CARtAgO* e *Moise*.

O modelo *Moise* é implementado pelo *JaCaMo* através da infraestrutura organizacional ORA4MAS (HÜBNER et al., 2009), onde a ideia básica é criar organizações em termos de agentes e artefatos, tendo como base o meta-modelo A&A apresentado na seção 2.3.2.1, e implementado através do arcabouço *CARtAgO*.

Assim, no *JaCaMo* a organização é composta por agentes organizacionais,

responsáveis por gerenciar a organização, e artefatos organizacionais. Esses artefatos são usados intencionalmente de um lado por agentes que participam da organização, a fim de tornar suas atividades mais eficazes e, do outro lado, por agentes organizacionais para controlar, gerenciar e modificar a organização em tempo de execução (HÜBNER et al., 2009).

Para a implementação de organizações com base no modelo *Moise*, dois são os artefatos organizacionais presentes no *JaCaMo* (BOISSIER et al., 2011):

- *GroupBoard*: artefato utilizado para gerenciar o ciclo de vida de grupos específicos de agentes; por exemplo, se um agente escolhe adotar um papel em um determinado grupo, este deverá executar a operação *adoptRole* do *GroupBoard* que representa o grupo desejado.
- *SchemeBoard*: artefato utilizado para gerenciar a execução de um esquema social; por exemplo, um agente pode se comprometer com uma missão ou dizer à organização que alcançou um dos objetivos que foi atribuído a ele, através da execução das ações *commitMission* e *setGoalAchieved* do *SchemeBoard* que representa o esquema social desejado.

Além das operações, os artefatos possuem propriedades observáveis que tornam visível aos agentes o estado da organização. Por exemplo, o artefato *GroupBoard* conta com uma propriedade observável acerca da disponibilidade de papéis em um grupo. Assim, agentes podem descobrir quais são os papéis existentes, raciocinar sobre eles e, de modo autônomo, decidir se adotam ou não um determinado papel.

Portanto, um SMA desenvolvido através do arcabouço *JaCaMo* é composto por uma organização distribuída em um ou vários *workspaces*, rodando em um ou vários nós da rede (Figura 12). Agentes rodando em um nó podem entrar e trabalhar concorrentemente em vários *workspaces* (incluindo os remotos).

Desta forma, o *JaCaMo* integra e estende as três tecnologias apresentadas, definindo ligações entre os conceitos das diferentes dimensões (agente, ambiente e organização), obtendo um modelo de programação uniforme e consistente que simplifica a combinação dessas dimensões no desenvolvimento de SMA.

Figura 12: Visão geral de um SMA *JaCaMo*, destacando suas três dimensões (BOISSIER et al., 2011).

2.4 Considerações Finais

Neste capítulo, foram apresentados os conceitos básicos para a construção de SMA: agentes, ambiente, interação e organização. Além disso, discutimos a importância da organização em SMA, e as vantagens de se adotar uma abordagem de desenvolvimento com foco na organização, em comparação com a abordagem clássica focada no agente. Por fim, discutimos a programação de SMA utilizando o arcabouço *JaCaMo*, o qual será utilizado neste trabalho de pesquisa.

3 *MULTI-AGENT PROGRAMMING CONTEST*

O MAPC¹ é uma competição realizada todos os anos com o propósito de estimular a pesquisa no campo da programação de SMA (BEHRENS; KÖSTER; SCHLESINGER, 2012). Em cada rodada do evento, dois times de agentes são situados no mesmo ambiente e competem diretamente num cenário definido pelos organizadores. Por se tratar de uma competição direta, é um cenário interessante para avaliar e comparar diferentes sistemas, identificar pontos fortes e fracos, e promover o desenvolvimento de todos os participantes.

Em 2011, o MAPC definiu como tema para a competição o cenário “*Agents on Mars*”, onde o principal objetivo é controlar zonas de uma mapa colocando os agentes em posições apropriadas. Um resumo deste cenário é apresentado a seguir.

3.1 *Cenário Agents on Mars*

No cenário “*Agents on Mars*”, dois times de agentes competem para dominar os melhores poços de água e as melhores zonas de Marte. O ambiente é representado por um grafo ponderado, em que os vértices denotam poços e possíveis localizações para os agentes e as arestas indicam a possibilidade de atravessar de um vértice para outro, com um custo em energia para o agente. Cada vértice possui um valor correspondente ao poço de água nele localizado, e esse valor é utilizado para o cálculo do valor das zonas ocupadas pelos agentes.

Uma zona é um sub-grafo coberto por uma equipe de acordo com um algoritmo de coloração baseado na noção de domínio (BEHRENS; KÖSTER; SCHLESINGER, 2012). Sendo V o conjunto de vértices, E o conjunto de arestas, ag o conjunto de

¹<http://multiagentcontest.org/>.

agentes e T o conjunto de equipes, temos que a coloração do grafo é dada pelo mapa:

$$c : V \rightarrow T \cup \{none\} \quad (3.1)$$

Considera-se que um vértice v é colorido se $c(v) \neq none$, com $v \in V$. Dado $ag(v)$ igual ao número de agentes no vértice v , temos que a coloração é determinada pelo seguinte algoritmo:

1. Primeiramente, colorem-se os vértices em que existem agentes sobre eles. Assim, temos que $c(v) = t$ se $ag(v) \neq 0$, onde t é o time que domina o vértice v . Um vértice v é dominado por t se este possui o maior número de agentes em v . Se nenhum time domina o vértice, então $c(v) = none$;
2. Em seguida, a coloração é estendida para vértices vazios, vizinhos diretos dos vértices dominados. Formalmente, $c(v) = t$ se $ag(v) = 0$, com t igual ao time que domina a maioria dos vizinhos do vértice que está sendo colorido. Neste caso, um time precisa dominar ao menos dois vértices vizinhos para que um vértice vazio herde a sua cor;
3. Alguns dos vértices coloridos com a cor de um time t podem representar uma *fronteira* que isola uma parte do grafo de todos os agentes dos outros times. Um vértice v foi isolado por uma equipe t se para todos os agentes ag pertencentes a um time t' , onde $t' \neq t$, não existe um caminho de ag_n para v que não inclua um vértice v' tal que $c(v') = t$;
4. Por fim, $c(v) = none$ se nenhuma das condições anteriores foram satisfeitas.

Este algoritmo permite que os times consigam colorir/ocupar sub-grafos em que o número de vértices é maior que o número de agentes. A Figura 13 mostra parte de um mapa com os sub-grafos coloridos.

O mapa é desconhecido dos agentes no começo da simulação. Assim, cada equipe precisa explorar o grafo antes de começar a conquistar as zonas, dado ainda que os agentes possuem visão limitada do mapa e só recebem percepções sobre vértices próximos. Além disso, às vezes, uma equipe precisa sabotar o outro time para conseguir aumentar sua área, ou se defender para não perder zonas para o oponente.

Figura 13: Cenário “Agents on Mars”.

Cada time é formado por 28 agentes² que podem ser de um dos cinco possíveis tipos. Os tipos, descritos na Tabela 2, definem características próprias do agente, tais como nível de vida, energia máxima, força e visibilidade, e as ações que o agente pode realizar no ambiente. Por exemplo, os exploradores podem encontrar poços de água e ajudar a explorar o mapa, os sentinelas possuem sensores de longa distância e assim podem observar grandes áreas, os sabotadores podem atacar e desativar os inimigos, os inspetores podem espionar os adversários, e os reparadores podem consertar agentes danificados. Cada time é formado por 6 exploradores, 6 inspetores, 6 reparadores, 6 sentinelas e 4 sabotadores.

Se uma equipe atinge um marco importante, ele recebe uma recompensa em dinheiro. A recompensa ganha por uma equipe pode ser usada para equipar os agentes, aumentando, por exemplo, o máximo de energia ou a força de um agente. Existem diferentes marcos que podem ser atingidos durante uma competição, tais como ter zonas com valores fixos (por exemplo, 10 ou 20), número fixo de ataques bem sucedidos, ou número fixo de defesas realizadas com sucesso. Se não for usado, o dinheiro ganho é somado à pontuação total da equipe.

O objetivo do jogo é maximizar a pontuação do time, que é definida como a soma dos pontos obtidos pelas zonas ocupadas com o dinheiro ganho (e não gasto) em cada

²O número de agentes por time passou de 20 para 28 em 2013.

passo da simulação, como mostra a Equação 3.2:

$$pontuacao = \sum_{p=1}^{passos} (zonas_p + dinheiro_p) \quad (3.2)$$

Tabela 2: Papéis e ações.

	explorador	reparador	sabotador	sentinela	inspetor
recarregar	X	X	X	X	X
atacar			X		
defender		X	X	X	
mover	X	X	X	X	X
sondar ³	X				
examinar ⁴	X	X	X	X	X
inspecionar ⁵					X
comprar	X	X	X	X	X
reparar		X			

Os agentes de cada time rodam localmente (nos computadores dos participantes), enquanto o ambiente simulado, no qual os times competem, é executado em um servidor remoto, pertencente a organização da competição. A comunicação entre agentes e o servidor remoto é feita através de sessões TCP/IP onde mensagens XML são trocadas. Em cada passo da simulação o agente recebe informações sobre o estado do ambiente (vértices dominados, inimigos próximos, nível de vida, etc) e envia para o servidor do MAPC a ação que deseja realizar.

Todos os times jogam 3 vezes contra cada time adversário e aquele que ganhar mais partidas vence a competição. Em cada partida, um novo mapa é gerado aleatoriamente, alterando-se assim o número e valor de vértices, arestas e, conseqüentemente, o número, valor e posição das boas regiões no grafo.

³Inicialmente, os agentes não tem conhecimento sobre qual é o valor dos poços de água. Apenas após sondar um vértice é que o time toma conhecimento do valor do poço.

⁴A priori, os agentes não conhecem qual o custo de se atravessar de um vértice a outro. Para descobrir qual o valor de cada aresta, o agente precisa examiná-la antes.

⁵Esta ação coleta informações sobre os oponentes presentes em vértices vizinhos, tais como nível de vida, energia e papel.

3.2 *LTI-USP Team*

Para participação nas edições de 2012 e 2013 do MAPC, foi desenvolvido através do arcabouço *JaCaMo* um SMA chamado de “*LTI-USP Team*” (FRANCO; ROSSET; SICHMAN, 2013; FRANCO; SICHMAN, 2013). A arquitetura implementada, assim como as estratégias organizacionais utilizadas são descritas a seguir.

3.2.1 Arquitetura

A arquitetura do SMA é apresentada na Figura 14 e baseia-se no modelo BDI. Os agentes foram desenvolvidos através da plataforma *Jason*, e cada agente possui sua própria *thread* de controle, planos, uma base de crenças e modelo de mundo. Os planos são especificados em *AgentSpeak*, e em cada passo da simulação o agente decide qual plano será executado de acordo com as suas crenças e visão local do mundo.

O modelo de mundo consiste de um grafo desenvolvido em Java utilizando classes e estruturas de dados simples. Este modelo capta todos os detalhes recebidos do servidor do MAPC, tais como vértices e arestas exploradas, posição dos adversários, companheiros de equipe desativados, etc. Em cada passo da simulação, o modelo de mundo do agente é atualizado com as percepções recebidas do servidor e com as informações recebidas dos outros agentes. O agente pode acessar ou alterar o estado do seu modelo de mundo através de ações internas desenvolvidas em Java. Alguns exemplos de ações internas são:

- *closer_repairer(Pos)*, que devolve ao agente a posição do agente reparador mais próximo,
- *move_to_target(Pos, Target, NextPos)*, que diz ao agente para qual vértice ele deve se mover a fim de alcançar determinada posição no grafo.

Algumas percepções recebidas do servidor são também armazenadas na base de crenças do agente, tais como seu papel, seu nível de energia e sua posição. Isto permite que o agente tenha acesso direto a essas informações sem ter que acessar o modelo de mundo. As percepções sobre vértices, arestas e outros agentes não são armazenadas na base de crenças para não comprometer o desempenho do agente, uma vez que poderia ser muito caro atualizar e acessar a base de crenças com tanta informação: o mapa de Marte utilizado na competição continha cerca de 400 vértices e centenas de arestas.

Figura 14: Arquitetura do "LTI-USP Team".

A comunicação entre os agentes ocorre através do *Jason*. Para reduzir a sobrecarga de comunicação, os agentes transmitem aos outros apenas as percepções novas, isto é, apenas as percepções recebidas do servidor que produzem alguma atualização no seu modelo de mundo. Por esta razão, existe uma forte troca de informação entre os agentes no início de uma partida, devido à comunicação sobre percepções novas e especialmente aquelas relacionadas à estrutura do mapa, como vértices e arestas. No entanto, a sobrecarga de comunicação diminui à medida que os agentes vão construindo um modelo de mundo mais completo. Os agentes se comunicam para:

- (i) informar os outros agentes sobre a estrutura do mapa;
- (ii) informar sobre agentes inimigos;

- (iii) pedir que algum outro agente vá repará-lo;
- (iv) pedir que um agente vá para determinado vértice.

Os agentes se comunicam com o servidor da competição através da interface EIS-MASSim incluída no pacote de software do simulador distribuído aos participantes. O EISMASSim é baseado em um padrão proposto para a interação agente-ambiente, denominado EIS⁶ (BEHRENS; DIX; HINDRIKS, 2009). Tal padrão automaticamente estabelece e mantém conexões autenticadas com o servidor, além de abstrair a comunicação entre agente e servidor para simples chamadas de métodos em Java. A fim de utilizar essa interface, foi necessário estender a arquitetura padrão de agentes do *JaCaMo* para que fosse possível a um agente não só perceber e agir sobre os artefatos do *CARtAgO*, mas também interagir com o servidor.

3.2.2 Estratégias

As estratégias utilizadas pelo “*LTI-USP Team*” nas edições de 2012 e 2013 do MAPC são apresentadas a seguir.

3.2.2.1 MAPC 2012

Pode-se definir a estratégia adotada como uma combinação da estratégia organizacional, das estratégias específicas de cada papel, e da estratégia de coordenação.

- **Estratégia Organizacional**

A estratégia principal adota em 2012 foi dividir os agentes em três subgrupos: dois grupos responsáveis por ocupar as melhores zonas do mapa (subgrupos *zona1* e *zona2*), e um encarregado de sabotar o time inimigo (subgrupo *sabotagem*). Para organizar os agentes dessa maneira, foram utilizados os artefatos organizacionais fornecidos pelo *Moise*, e definidas as especificações estrutural (EE), funcional (EF) e deôntica (ED) do SMA.

Na EE foram definidos os três subgrupos citados anteriormente, e sete possíveis papéis: os cinco papéis especificados pelo cenário (explorador, sabotador, sentinela,

⁶Disponível em <http://sourceforge.net/projects/apleis/>.

reparador e inspetor), e mais dois papéis chamados de “marciano” e “coordenador” (Figura 15). O coordenador é o responsável em conduzir os agentes a ocuparem as melhores zonas de Marte e, diferente dos outros agentes, o coordenador não se comunica com o servidor do MAPC, sendo um agente interno do sistema. O marciano é o papel padrão adotado pelos outros agentes no início de uma partida, enquanto não recebem do servidor a informação sobre qual papel desempenhar.

Figura 15: Especificação Estrutural do “LTI-USP Team” 2012.

Cada subgrupo possui um meta global associada, definida na EF. Na Figura 16, essas metas globais são representadas como raiz das árvores, que apresentam em suas folhas as submetas à serem atingidas pelos agentes. O rótulo que aparece logo acima de uma meta representa a missão à qual o agente deve estar comprometido, a fim de atingir a meta relacionada. Para cada uma das missões existe um plano que o agente deve seguir.

As diferentes metas relacionadas a cada subgrupo são apresentadas a seguir:

- **ocuparZona1:** Os agentes no grupo zona1 devem ocupar a melhor zona no grafo seguindo as orientações dadas pelo agente coordenador. Além disso, um dos exploradores fica responsável por sondar os vértices do grafo, a fim de encontrar as melhores zonas a serem ocupadas. Por fim, um inspetor tem a missão de identificar o papel de cada agente no time inimigo. Apenas ao tomar conheci-

Figura 16: Especificação Funcional do “LTI-USP Team” 2012.

mento do papel de todos os oponentes é que o inspetor se junta ao resto do time na missão de ocupar a melhor zona do grafo.

- **ocuparZona2:** Todos os agentes no grupo zona2 tem a missão de ocupar a segunda melhor zona no grafo, ou ajudar o grupo zona1 a formar uma área maior. Se este grupo deve ou não se juntar ao outro grupo é determinado pelo coordenador, que verifica a existência ou não de duas boas zonas dispersas no grafo.
- **sabotarOponente:** O grupo sabotagem é formado por um sabotador e um sentinela. A missão do sabotador é atacar os oponentes que ocupam bons vértices; e a missão do sentinela é de ajudar na sabotagem, movendo-se para dentro da zona do time inimigo de forma a quebrá-la ou fazê-la diminuir.

Por fim, a ED representada na Tabela 4 define quais papéis tem a obrigação ou permissão de se comprometer com cada missão.

Tabela 4: Especificação Deontica do “LTI-USP Team” 2012.

Papel	Missão	Relação Deontica
explorador	mExplorar, mOcuparZona1	permissão
explorador	mOcuparZona2	obrigação
reparador	mRepararZona1, mRepararZona2	obrigação
sabotador	mSabotar, mOcuparZona1, mOcuparZona2	obrigação
sentinela	mAjudarSabotar, mOcuparZona1, mOcuparZona2	obrigação
inspetor	mInspeccionar, mOcuparZona1	permissão
inspetor	mOcuparZona2	obrigação
coordenador	mCoordenar	obrigação

• Estratégias Dependentes do Papel

Para cada papel, foram desenvolvidas estratégias específicas. Por exemplo, os exploradores devem sondar e examinar todo vértice ou aresta no seu caminho, enquanto os inspetores devem sempre procurar inspecionar inimigos próximos.

Os agentes também se comportam diferentemente, dependendo do papel que desempenham, quando encontram um oponente no mesmo vértice. Os sabotadores sempre atacam o oponente, enquanto os sentinelas procuram se defender e os outros agentes fogem para um outro vértice.

• Estratégia de Coordenação

A estratégia adotada foi baseada na centralização da coordenação, isto é, um único agente (coordenador) é responsável por determinar quais as melhores zonas do mapa e, em seguida, conduzir os outros agentes a ocuparem essas zonas. A escolha da coordenação centralizada foi feita para permitir o rápido desenvolvimento da equipe, uma vez que a motivação principal em nossa primeira participação no MAPC era se concentrar no uso da plataforma *JaCaMo* e não nos aspectos de coordenação.

Desta forma, o coordenador constrói o seu modelo de mundo através das percepções difundidas pelos outros agentes e, sempre que o modelo de mundo é atualizado, calcula quais são as duas melhores zonas no gráfico. O coordenador então pede para que os agentes (de cada um dos dois grupos responsável por ocupar as zonas) se movam para dentro das zonas calculadas. Quando todos os agentes estão dentro das zonas, o coordenador passa então a olhar nos vértices vizinhos a estas, procurando posições para as quais os agentes possam se mover a fim de aumentar o tamanho da área conquistada.

A melhor zona é obtida calculando-se para cada vértice a soma do seu valor com a dos seus vizinhos diretos e de segundo grau. O vértice com a maior soma de valores é

o centro da melhor zona, sendo que zonas com valor inferior a 10 são desconsideradas. O mesmo cálculo é feito para determinar qual a segunda melhor zona no mapa, mas dessa vez desconsiderando os vértices pertencentes a melhor zona obtida até então. Se duas zonas são encontradas, o coordenador as designa aos subgrupos zona1 e zona2. Caso contrário, a mesma zona será atribuída aos dois subgrupos.

3.2.2.2 MAPC 2013

Para a edição de 2013, alterou-se substancialmente a organização do time, adicionando diferentes papéis aos agentes, como mostrado na Figura 17.

Figura 17: Especificação Estrutural do “LTI-USP Team” 2013.

Foi mantida a estratégia de distribuir os agentes em três subgrupos (agora chamados de pelotão1, pelotão2 e infantaria), no entanto, em relação aos papéis

dos agentes, decidiu-se por não mapear os cinco tipos de agentes especificados no cenário (explorador, inspetor, reparador, sabotador e sentinela) diretamente para papéis no “LTI-USP Team”. Ao invés disso, de acordo com a estratégia adotada, definiu-se diferentes papéis adicionais para o SMA. Cada um desses papéis tem uma missão associada, podendo ser adotados por um ou mais tipos de agentes. Por exemplo, o papel de `soldado_explorador` pode ser adotado apenas por exploradores, enquanto o papel de `soldado` pode ser assumido por todos os tipos de agentes. A seguir descrevemos as missões relacionadas a cada papel.

- **explorador** (explorador): Explora o grafo sondando cada vértice e examinando todas as arestas em seu caminho;
- **explorador_ajudante** (explorador): Ajuda o `explorador` a explorar o mapa, mas apenas nos primeiros 250 passos da simulação. Após isso, o agente abandona este papel para adotar o papel de `soldado` no subgrupo `pelotão1`;
- **soldado** (todos os tipos): Tentam ocupar uma das melhores zonas indicadas pelo coordenador. Quando todos os vértices da zona atribuída são ocupados, o `soldado` começa a procurar por vértices vizinhos aos quais ele poderia se mover, de forma a aumentar o tamanho da zona;
- **guarda** (sabotador): Defendem a zona ocupada atacando qualquer oponente que esteja próximo a ela, ou tentando invadi-la;
- **médico** (reparador): Ocupa o centro da melhor zona designada e é responsável por reparar os agentes no subgrupo, ou outros agentes que eventualmente precisem ser reparados, tal como o `explorador_ajudante`. No “LTI-USP Team”, os agentes danificados se movem até os reparadores para serem reparados;
- **soldado_explorador** (explorador): Explora a zona do time sondando cada vértice cujo valor é desconhecido. Quando todos os vértices são sondados, o `soldado_explorador` ajuda os `soldados` a aumentar o tamanho da área ocupada;
- **sabotador** (sabotador): Ataca qualquer oponente próximo, ou o oponente que ocupa o melhor vértice;
- **sentinela** (sentinela): Tenta sabotar o oponente se movendo para dentro da sua zona;

- **reparador** (reparador): Segue o **sabotador**, mas sempre se mantendo a dois vértices de distância, de forma a estar preparado a reparar o sabotador quando necessário, mas sem se ariscar muito;
- **coordenador** (nenhum): Responsável por informar ao agentes quais são as melhores áreas no mapa.

Em contraste com o time do ano anterior, alterou-se o SMA de forma que este fosse mais descentralizado. O coordenador continua responsável por determinar quais são as melhores zonas do mapa, mas no entanto cada agente decide por si só qual vértice vazio irá ocupar para formar uma zona ou aumentá-la.

3.3 Resultados

A seguir são apresentados os resultados obtidos em cada uma das edições do MAPC.

3.3.1 MAPC 2012

A participação no MAPC 2012 foi de grande valia no objetivo de conhecer melhor o funcionamento do arcabouço *JaCaMo*. Apesar dos esforços não terem sido direcionados na procura de uma melhor estratégia, o time desenvolvido acabou o torneio em quarto lugar em um total de sete equipes, como mostrado na Tabela 5⁷.

Tabela 5: Resultado final do MAPC 2012.

Posição	Time	Filiação	País	Pontos
1	UFSC	Universidade Federal de Santa Catarina	Brasil	51
2	Python-DTU	Technical University of Denmark	Dinamarca	48
3	TUB	Technical University of Berlin	Alemanha	30
4	LTI-USP	Universidade de São Paulo	Brasil	27
5	AiWYX	Sun Yat-Sen University	China	24
6	PGIM	Islamic Azad University of Malayer	Irã	9
7	Streett	-	EUA	0

Durante o desenvolvimento do time de agentes, o arcabouço *JaCaMo* provou ser bastante completo, fornecendo todas as ferramentas necessárias para desenvolver o SMA. Além disso, o uso do modelo organizacional *Moise* ajudou no projeto da equipe,

⁷Fonte: <http://multiagentcontest.org/2012>.

uma vez que ele fornece uma forma de estruturar o SMA, não somente através da especificação de grupos e papéis, mas principalmente na definição de como os objetivos globais deverão ser alcançados.

O grande obstáculo no desenvolvimento do time foi lidar com os problemas de desempenho relacionados com o uso concorrente dos artefatos organizacionais.

No início de cada partida, os agentes recebem quase que ao mesmo tempo do servidor do MAPC o papel que irão desempenhar naquela partida para então tentar entrar em um dos três grupos especificados (*zona1*, *zona2* e *sabotagem*). Ao fazerem isso, é possível que, por exemplo, todos os quatro sabotadores queiram entrar no grupo *sabotagem*, mas, como mostrado na Figura 15, neste grupo só é permitido um sabotador. Neste caso, três sabotadores serão impedidos de entrar no grupo *sabotagem* e terão que tentar se juntar a outro grupo. Em testes realizados antes da competição, foi notado que a chamada de uma ação organizacional, tais como *adoptRole* ou *commitMission*, é muito cara, e que o número de tentativas feitas por um agente até finalmente conseguir entrar em um grupo podia ser muito alto. Isto fazia com que alguns agentes perdessem alguns passos no início da simulação, já que em cada passo os agentes tem um prazo limitado (1000 ms) para enviarem as suas ações.

Para corrigir esse problema, passou-se para o agente coordenador a responsabilidade de distribuir entre os outros agentes os grupos e missões. Desta forma, foi evitada a ocorrência de colisões, como agentes tentando entrar no mesmo grupo ou se comprometer com a mesma missão, melhorando assim o desempenho do SMA.

Após o término da competição foram realizados alguns experimentos para avaliar de forma preliminar o desempenho da operação *adoptRole*, em função do número de agentes acessando a operação em concorrência. Os resultados dessa avaliação de desempenho foram apresentados num artigo publicado na sétima edição do “Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações” (WE-SAAC) (FRANCO; SICHMAN, 2013).

3.3.2 MAPC 2013

Cinco equipes participaram novamente da competição em 2013, na qual o “*LTI-USP Team*” terminou em terceiro lugar (Tabela 6⁸).

⁸Fonte: <http://multiagentcontest.org/2013>.

Tabela 6: Resultado final do MAPC 2013.

Posição	Time	País	Pontos
1	UFSC	Brasil	36
2	Python-DTU	Dinamarca	27
3	LTI-USP	Brasil	15
4	TUB	Alemanha	6
5	AiWYX	China	6

Devido à modularidade do arcabouço *JaCaMo*, não foi complicado alterar o time para a competição. Foi possível reusar toda a arquitetura construída em 2012 para comunicação com o servidor do MAPC e para o funcionamento dos agentes. As principais alterações consistiram na organização do time, através das especificações do *Moise*, e a criação dos planos para os novos papéis. Desta forma, o *JaCaMo* provou ser uma plataforma muito flexível, permitindo alterar facilmente a estratégia adotada e testar algumas variações.

Comparando com o ano anterior, foi possível obter uma melhor estabilidade em relação às zonas ocupadas. Isso foi possível graças: (i) a mudança para uma abordagem mais descentralizada, com os agentes decidindo por si só para onde se mover a fim de criar ou expandir a área do time; e (ii) a uma estratégia mais defensiva em que cada pelotão dispõe de um guarda pronto a atacar qualquer oponente próximo, e um médico no centro da zona focada em reparar rapidamente os agentes.

3.4 Considerações Finais

Através da participação no MAPC foi possível conhecer melhor o funcionamento do arcabouço *JaCaMo* e aprimorar nosso conhecimento sobre o desenvolvimento de SMA.

Durante a competição, diferentes estratégias foram utilizadas pelos times participantes, que conduziram a adoção de diferentes estruturas organizacionais: com mais ou menos agentes responsáveis por atacar o oponente, com os agentes divididos em dois ou mais grupos, entre outras possíveis variações.

Como o desempenho de um time é diretamente ligado às estratégias adotadas, este também é influenciado pela organização dos agentes. Desta forma, o MAPC fornece um cenário interessante onde é possível comparar não apenas times, tecnologias e estratégias, mas também organizações.

4 AVALIAÇÃO ORGANIZACIONAL EM SMA

Como vimos na seção 2.2, a organização é um componente fundamental para restringir o comportamento dos agentes, a fim de aumentar a eficiência global do sistema. Entretanto, encontrar a estrutura organizacional ótima, que melhor resolve determinado problema, é uma questão sem resposta. Isto porque o desempenho de uma organização depende de vários fatores, tais como:

- característica da(s) tarefa(s) a ser(em) desempenhada(s): subdivisão, dependências;
- características dos agentes: capacidades cognitivas, inteligência, limitações;
- ambiente: dinamismo e incertezas;
- resultados desejados: eficiência, precisão, custo mínimo.

Assim, dado essa variedade de fatores, é de consenso entre os pesquisadores que não há estrutura organizacional única capaz de apresentar o melhor desempenho em todas as possíveis situações (CARLEY; GASSER, 1999; HORLING; LESSER, 2004).

De fato, um dos principais problemas na área de SMA é determinar que estruturas organizacionais adotar para cada tipo de problema, e quais os custos e benefícios de diferentes configurações organizacionais.

4.1 Avaliação Organizacional

Avaliar o desempenho de uma organização é um trabalho difícil e complexo. Frequentemente o espaço de parâmetros dos conjuntos de opções é tão extenso que o pesquisador não consegue explorar todas as possíveis combinações, para assim conseguir determinar por completo o desempenho do sistema (CARLEY; GASSER, 1999).

Para lidar com esse problema, pesquisadores normalmente realizam um conjunto de experimentos, a fim de responder questões específicas, tais como: qual o impacto do número de níveis em uma hierarquia para a conclusão de determinada tarefa; qual a melhor forma de divisão das tarefas entre os agentes; qual o número mínimo de agentes necessários para o sistema alcançar um nível satisfatório de desempenho; entre outras.

Cada experimento consiste de uma série de simulações onde um pequeno conjunto de parâmetros varia sistematicamente entre uma faixa de valores. Todos os outros parâmetros são normalmente mantidos constantes ou autorizados a variar randomicamente. Ao final, aplica-se algum teste estatístico, a fim de analisar os dados obtidos e obter alguma resposta às questões relacionadas ao experimento.

Como exemplo de avaliação organizacional podemos citar o trabalho de Bryan Horling e Victor Lesser. Em (HORLING; LESSER, 2005) os autores descrevem um SMA utilizado para enfrentar os desafios colocados por um ambiente de rede de sensores distribuídos. Neste sistema, a organização define como os agentes agem e interagem através da especificação dos possíveis papéis para os agentes e da divisão destes em setores responsáveis por monitorar diferentes regiões do ambiente. O artigo descreve a arquitetura do sistema em detalhes e apresenta resultados empíricos demonstrando que o desempenho de um sistema pode variar muito dependendo dos parâmetros organizacionais escolhidos, tais como neste caso o tamanho dos setores.

4.2 Métricas

Dependendo do objetivo e características desejadas no sistema, diferentes métricas podem ser utilizadas para medir o desempenho de uma organização. Assim, para diferentes problemas, pode-se avaliar a organização em termos de escalabilidade, confiabilidade, eficiência, entre outras. Em (HORLING; LESSER, 2005), por exemplo, o desempenho da organização é medida em termos da quantidade total de mensagens trocadas entre os agentes.

Outras métricas utilizadas na avaliação de organizações são apresentadas a seguir:

- eficiência na realização das tarefas e objetivos;
- ganho ou utilidade obtida pela organização ou pelos agentes;
- utilização de recursos;

- tempo necessário para se alcançar um objetivo.

Neste trabalho, para avaliar o desempenho dos times que competem no MAPC, será utilizada como métrica de desempenho a pontuação obtida por cada time, em cada partida.

4.3 Testes Estatísticos

Ao realizar um determinado experimento é muito importante definir corretamente a quantidade de simulações a serem realizadas. O número de simulações deve ser grande o suficiente para revelar as tendências no comportamento do sistema. Poucas simulações podem não produzir resultados representativos, podendo assim ser necessário realizar alguma experimentação inicial para determinar qual o número de simulações capazes de produzir resultados estáveis (HARRISON et al., 2007).

Normalmente, se aplica o seguinte procedimento para determinar o nível de precisão dos resultados obtidos em um experimento:

1. Executa-se um número N_0 de simulações;
2. Calcula-se a variância S_0^2 e o desvio padrão S_0 ;
3. O erro máximo E_0 associado a média obtida pode então ser calculado através da Equação 4.1, onde $z_{\alpha/2}$ corresponde ao desvio padrão normalizado para um intervalo de confiança de $(1 - \alpha)$:

$$E_0 = z_{\alpha/2} \cdot \frac{S_0}{\sqrt{N_0}} \quad (4.1)$$

4. Caso o valor E_0 calculado esteja fora do desejado, é possível utilizar a mesma equação para determinar o número de simulações N necessárias para se obter o nível de precisão E desejado. Assim, resolvendo a equação anterior para N temos:

$$N = \left(\frac{z_{\alpha/2} \cdot S_0}{E} \right)^2 \quad (4.2)$$

Em alguns experimentos, ao invés de estimar o valor obtido pela organização para determinada métrica de desempenho, queremos verificar se uma declaração sobre os

resultados é verdadeira ou falsa. Nestes casos, se faz uso de algum teste de hipótese (JOHNSON, 2000).

Ao realizar um teste de hipóteses o objetivo é validar ou não determinadas hipóteses sobre a população, com base na informação amostral, ou seja, o resultado das simulações. Os testes de hipóteses são sempre constituídos por duas hipóteses, a hipótese nula H_0 e a hipótese alternativa H_1 :

- Hipótese nula (H_0): é a hipótese que traduz a ausência do efeito que se quer verificar. Frequentemente, a hipótese nula consiste em afirmar que os parâmetros ou características matemáticas de duas ou mais populações são idênticos;
- Hipótese alternativa (H_1): é a hipótese que o investigador quer verificar. A hipótese alternativa deve ser contrária, oposta, antagônica à hipótese nula.

Num teste de hipóteses, o objetivo é o de tomar uma decisão, baseada na rejeição ou não da hipótese nula. Esta decisão é baseada na informação disponível, obtida a partir da amostra recolhida da população. No caso de se rejeitar H_0 , aceita-se como válida a hipótese alternativa H_1 .

Um risco está sempre associado a uma decisão: o de tomar uma decisão errada. No caso dos testes de hipóteses, a tomada de decisão para a população é baseada na informação amostral, e dependendo da amostra escolhida pode-se cometer erros. Assim, em todo teste de hipótese se estabelece um nível de significância α que representa a probabilidade de rejeitar a hipótese nula quando ela é efetivamente verdadeira.

O valor da probabilidade de se obter o efeito observado, dado que a hipótese nula é verdadeira, é chamado de valor-p. Se o valor-p for menor que o nível de significância α estipulado, rejeita-se a hipótese nula. Ao contrário, se o valor-p for maior, aceita-se a hipótese nula. Quanto mais baixo o valor-p, maior é a evidência contra a hipótese nula.

Dentre os diversos testes de hipótese, é importante identificar aqueles que são ou não aplicáveis a cada situação. Para realizar as escolhas adequadas, deve-se considerar alguns parâmetros básicos dos dados a serem analisados, tais como (TRIOLA, 2008):

- Número de amostras: O número de grupos distintos sendo analisados;

- Relações entre amostras: Amostras podem ser pareadas ou não-pareadas. Amostras pareadas ocorrem quando cada observação na primeira amostra tem relação com uma observação na segunda amostra.
- Escala numérica: A forma como os dados foram registrados (escala qualitativa, quantitativa discreta e quantitativa contínua);
- Distribuição: A densidade de probabilidade (distribuição de probabilidade) dos dados (normal ou não-normal);

A Tabela 7 abaixo indica as técnicas estatísticas que podem ser aplicadas para a comparação entre as amostras de dois ou mais grupos de dados.

Tabela 7: Testes estatísticos (TRIOLA, 2008).

Nº de Amostras	Tipo de Relação	Distribuição	Escala Numérica	Análises Aplicáveis
2	Não-Pareadas	Normal	contínua, discreta	<i>t</i> de Student
		Não-Normal	contínua, discreta	Mann-Whitney U
			qualitativa	Qui-Quadrado
	Pareadas	Normal	contínua, discreta	<i>t</i> de Student Pareado
		Não-Normal	contínua, discreta	Wilcoxon Pareado
			binária	McNemar
≥3	Não-Pareadas	Normal	contínua, discreta	ANOVA
		Não-Normal	contínua, discreta	Kruskal-Wallis
			qualitativa	Qui-Quadrado
	Pareadas	Normal	contínua, discreta	ANOVA
		Não-Normal	contínua, discreta	Friedman
			qualitativa	Q de Cochran

Neste trabalho foi utilizado o teste de “Wilcoxon Pareado” (*Wilcoxon signed-rank test*), cuja escolha baseou-se no fato de desconhecermos a priori se os resultados obtidos seguiriam uma distribuição normal, e devido as amostras serem pareadas, já que a pontuação final de um time tem relação com a pontuação do outro time em uma mesma partida/simulação.

Para o cálculo do Teste de “Wilcoxon Pareado”, considere dois conjuntos de valores dependentes de tamanho n , denominados X e Y , a serem comparados. Como as observações são pareadas, podemos considerar que temos na realidade uma amostra de pares:

$$\{(X_1, Y_1), \dots, (X_n, Y_n)\}$$

Definimos $D_i = X_i - Y_i$, para $i = 1, 2, \dots, n$; obtendo assim a amostra resultante das diferenças entre os valores de cada par:

$$\{D_1, D_2, \dots, D_n\}$$

Em seguida, os valores D_i são ordenados do menor para o maior valor, e para cada um é atribuído um número de posição P_i . Assim, ao menor valor atribui-se o número de posição 1, ao segundo menor valor atribui-se o número de posição 2 e assim sucessivamente.

Finalizada a atribuição do número das posições, realiza-se a sua somatória, conforme a Equação 4.3:

$$W = \left| \sum_{i=1}^n [sgn(D_i) \cdot P_i] \right| \quad (4.3)$$

onde, sgn é a função sinal:

$$sgn(x) = \begin{cases} -1 & : x < 0 \\ 0 & : x = 0 \\ 1 & : x > 0 \end{cases} \quad (4.4)$$

A partir do valor de W , calcula-se o valor de z para determinar se a diferenças entre os dois grupos é estatisticamente significativa. Para $n \geq 10$, z é calculado pela expressão:

$$z = \frac{W - 0.5}{\sigma_W} \quad (4.5)$$

onde σ_W representa o desvio padrão, calculado pela seguinte equação:

$$\sigma_W = \sqrt{\frac{n(n+1)(2n+1)}{6}} \quad (4.6)$$

Utiliza-se então o valor calculado de z para obter o valor-p através da tabela de distribuição normal. Caso o valor obtido de p seja menor ou igual ao valor de significância (α) estipulado, H_0 é rejeitado, aceitando-se H_1 . Em caso contrário, H_0 não é rejeitado.

4.4 Metodologia

Por fim, de acordo com o apresentado nas seções anteriores, definiu-se uma metodologia para avaliação organizacional de SMA, que foi utilizada neste trabalho, e cujos passos são os seguintes:

1. Escolher os parâmetros organizacionais a serem avaliados: número de agentes, divisão das tarefas, etc;
2. Definir como as características do ambiente serão alteradas: irão se manter constantes, irão variar dentro de uma faixa de valores pré-definida, ou poderão variar aleatoriamente;
3. Definir a métrica de desempenho a ser utilizada: eficiência, utilização de recursos, tempo, etc;
4. Definir os experimentos: número de simulações, coleta dos dados, etc;
5. Executar os experimentos;
6. Realizar a análise estatística dos resultados: média dos valores obtidos, teste de hipótese.

4.5 Considerações Finais

Comparar organizações ou conhecer a influência que determinado parâmetro organizacional pode ter no sistema é importante para a escolha da organização a ser adotada em um SMA.

Como apresentado neste capítulo, a avaliação de uma organização é normalmente feita através de simulações onde se procura verificar o efeito de determinado parâmetro organizacional no desempenho do sistema.

Dentro deste contexto, este trabalho tem como objetivo avaliar, através do resultado de diferentes simulações, qual o impacto de mudanças na organização no desempenho de um time que compete no MAPC. Além disso, também é avaliado o efeito de alterações em características do ambiente sobre o parâmetro organizacional estudado.

5 EXPERIMENTOS E RESULTADOS

Como visto na seção 2.2.2, a abordagem organizacional possui várias vantagens em relação à abordagem clássica de desenvolvimento de SMA, onde o foco são os agentes, principalmente no que diz respeito à obtenção de SMA mais eficientes. Por outro lado, como apresentado no capítulo 4, a organização é um dos fatores que possui grande influência no desempenho de SMA. Neste contexto, este trabalho consiste em realizar um estudo sobre o impacto da organização no desempenho de SMA.

Tal estudo pressupõem a escolha de um problema em um domínio de aplicação específico, cuja solução possa ser projetada e implementada mais facilmente através de um MAS centrado na organização. Um domínio adequado é o *Multi-Agent Programming Contest* (MAPC).

Desta forma, partindo do time desenvolvido para participação no MAPC 2013, vários experimentos foram realizados com o objetivo de identificar qual o impacto da estrutura organizacional no desempenho do SMA. No caso particular do MAPC, foi estudo qual o impacto do número de grupos no desempenho do time. Além disto, procurou-se avaliar também como variações no mapa utilizado podem influenciar o desempenho de cada SMA.

A seguir são apresentados detalhadamente os experimentos realizados, bem como uma análise dos resultados obtidos.

5.1 Experimentos

No MAPC, cada time enfrenta três vezes cada um dos outros times, e aquele que ganha o maior número de partidas vence a competição. Cada partida tem 750 passos e o mapa é gerado aleatoriamente. Assim, o que pode mudar de uma partida para a outra é o número de vértices, arestas e áreas de grande valor.

Este número variável de áreas de grande valor no mapa leva, em alguns casos, a situações em que ocupar uma única boa área é a melhor estratégia, enquanto em outros casos, é melhor dividir o time em grupos menores para tentar ganhar o controle sobre diversas áreas do grafo. Assim, realizamos diversos experimentos para analisar como o número de pelotões/grupos em nosso time pode impactar o seu desempenho global.

5.1.1 Variação de Parâmetros Organizacionais

Para os experimentos, sete times (TG1, TG2, TG3, TG4, TG6, TG8, e TG10) foram criados tendo como base o time desenvolvido para disputar o MAPC em 2013, mas cada um destes com um diferente número de pelotões, como mostrado na Figura 18.

Figura 18: Especificação estrutural dos times.

Cada time possui um grupo infantaria responsável por atacar o oponente, dois ex-

ploradores responsáveis por explorar o mapa, e o restante dos agentes foi dividido em pelotões conforme a Tabela 8. Esta divisão foi feita de forma que em cada pelotão tivéssemos praticamente o mesmo número de agentes, e respeitando a quantidade disponível de agentes de cada tipo ao atribuir os papéis. Por exemplo, cada time possui 4 agentes do tipo sabotador e um destes é utilizado no grupo infantaria. Desta forma, sobram 3 agentes sabotadores para serem distribuídos nos pelotões. Como o papel de guarda pode ser adotado apenas por agentes sabotadores, damos preferência para que os agentes assumam este papel.

Tabela 8: Divisão dos agentes em pelotões.

Time	Pelotões	soldado	guarda	médico	soldado_ explorador	Agentes
TG1	1	20	1	1	1	23
TG2	1	9	1	1	1	12
	2	8	1	1	1	11
TG3	1	5	1	1	1	8
	2	5	1	1	1	8
	3	4	1	1	1	7
TG4	1	3	1	1	1	6
	2	3	1	1	1	6
	3	3	1	1	1	6
	4	3	0	1	1	5
TG6	1	1	1	1	1	4
	2	1	1	1	1	4
	3	1	1	1	1	4
	4	2	0	1	1	4
	5	3	0	1	0	4
	6	3	0	0	0	3
TG8	1	1	1	1	0	3
	2	1	1	0	1	3
	3	1	0	1	1	3
	4	0	1	1	1	3
	5	1	0	1	1	3
	6	2	0	1	0	3
	7	3	0	0	0	3
	8	2	0	0	0	2
TG10	1	1	1	1	0	3
	2	1	1	0	1	3
	3	1	0	1	1	3
	4	0	1	1	0	2
	5	0	0	1	1	2
	6	1	0	1	0	2
	7	1	0	0	1	2
	8	2	0	0	0	2
	9	2	0	0	0	2
	10	2	0	0	0	2

Ao contrário do adotado no MAPC 2013, foram desconsiderados os vértices vizinhos de segundo grau para o cálculo das zonas a serem ocupadas pelos pelotões e nenhuma restrição foi imposta em relação ao valor mínimo das zonas a serem consideradas. Assim, cada pelotão recebe uma zona diferente a ser ocupada, mas os agentes são inteligentes a ponto de cooperar com pelotões vizinhos no sentido de criar uma zona maior.

Por fim, os experimentos consistiram em fazer os times competirem entre si em diferentes cenários/mapas. Em cada cenário os times enfrentaram 10 vezes cada um dos oponentes, e ao final de cada partida/simulação foram coletados dados sobre o vencedor e a pontuação final de cada time. Estas métricas serviram para indicar o desempenho de cada time.

5.1.2 Características do Ambiente

Para se analisar o impacto do ambiente sobre o desempenho das organizações, um total de 14 cenários foram utilizados, variando-se em cada um o número de áreas de grande valor, o tamanho destas e a distância entre elas. As “áreas de grande valor” foram consideradas como sendo os subgrafos formados por vértices de valor maior que 1, e cuja soma destes é maior que 10. Essas áreas são chamadas neste trabalho de *clusters*, e estes, dependendo do seu tamanho, podem ser divididos em uma ou mais zonas a serem conquistadas pelos pelotões.

Assim, em relação aos *clusters* as características que foram alteradas durante os experimentos foram as seguintes:

- *número*: valor numérico que representa a quantidade de áreas de grande valor (ou *clusters*) presentes no mapa;
- *homogeneidade*: valor binário (Sim ou Não) que representa se os *clusters* presentes no mapa possuem aproximadamente o mesmo valor;
- *dispersão*: valor binário (Sim ou Não) que representa a média das distâncias aproximadas entre os *clusters*.

Os cenários utilizados e suas características são apresentados na Tabela 9 e nas Figuras 19 e 20. Um maior detalhamento de cada cenário pode ser encontrado no Apêndice A.

Tabela 9: Características dos cenários.

Cenário	Clusters	Homogeneidade	Dispersão
SC1_P	1	-	-
SC1_G	1	-	-
SC2_SN	2	Sim	Não
SC2_SS	2	Sim	Sim
SC2_NS	2	Não	Sim
SC3_SS	3	Sim	Sim
SC3_SN	3	Sim	Não
SC4_SN	4	Sim	Não
SC4_SS	4	Sim	Sim
SC4_NN	4	Não	Não
SC5_NN	5	Não	Não
SC5_SS	5	Sim	Sim
SC6_SS	6	Sim	Sim
SC6_SN	6	Sim	Não

Figura 19: Cenários.

Note que cada *cluster*, dependendo do seu tamanho, pode vir a ser dividido em duas ou mais zonas a serem ocupadas pelos pelotões de um time.

Intuitivamente, nossa hipótese é a de que times com um maior número de pelotões sejam mais eficientes em mapas dispersos, homogêneos e com um maior número de *clusters*.

Figura 20: Cenários (continuação).

5.1.3 Execução dos Experimentos

Para executar os experimentos foram utilizadas 5 máquinas virtuais Linux, cada uma com 4GB de memória RAM e 2GHz de velocidade do processador. Uma das

máquinas virtuais era responsável por rodar duas instâncias do servidor do MAPC, enquanto os times eram processados nas outras máquinas.

Um *script* Python era responsável por executar os experimentos, inicializando remotamente o servidor e os times de agentes de acordo com os cenários a serem analisados. Após cada simulação, o *script* salvava os resultados, tais como a pontuação final dos times e os arquivos XML para visualização posterior das partidas.

Ao final das execuções, um outro *script* Python resumia os resultados, criando para cada cenário um arquivo CSV com a pontuação final obtida por cada time, em cada simulação.

Por fim, para realizar a análise estatística dos resultados, usamos um *script* R⁹, que consiste em uma linguagem de programação para cálculos estatísticos e gráficos. Para o cálculo do teste de hipótese de “Wilcoxon Pareado” utilizamos o método `wilcox.test()` do pacote padrão do R.

5.2 Metodologia de Análise

Como mostrado no capítulo 4, é importante escolher o método estatístico a ser utilizado para comparação das diferentes organizações. Neste trabalho utilizamos o teste de hipótese não-paramétrico denominado teste de “Wilcoxon Pareado”. Esta escolha baseou-se no fato de desconhecermos a priori se os resultados seguem uma distribuição normal, e devido as amostras serem pareadas, já que a pontuação final de um time tem relação com a pontuação do outro time em uma mesma partida/simulação. Mais detalhes sobre o teste de “Wilcoxon Pareado” podem ser encontrados na seção 4.3.

Para ilustrar o uso desta metodologia de análise estatística, tome-se o experimento realizado no cenário SC3_SS, um dos avaliados neste trabalho. Neste cenário, apresentado na Figura 21, existem 3 *clusters*, sendo 2 deles nos cantos inferiores e o outro no centro superior do mapa. Como já mencionado anteriormente, cada time enfrentou 10 vezes cada um dos oponentes. Em particular, no confronto entre TG2 e TG4, TG4 ganhou 8 das 10 simulações, conforme apresentado na Figura 22.

Este resultado é um bom indício de que para este cenário TG4 é melhor que TG2. Entretanto para confirmar isto, e descartar que ele tenha ocorrido aleatoriamente, é

⁹Disponível em <http://www.r-project.org/>

Figura 21: Cenário SC3_SS, contendo 3 *clusters*.

Figura 22: Cenário SC3_SS - Resultados das simulações TG2 vs TG4.

necessário aplicar um teste de hipótese sobre os resultados, a fim de validar possíveis conclusões. Assim, inicialmente, são definidas uma hipótese nula e uma hipótese alternativa adequadas ao contexto aqui estudado. Dado qualquer dois times A e B, podemos afirmar que:

- Hipótese nula (H_0): Não há diferença significativa entre o desempenho obtido por A com respeito ao desempenho de B;
- Hipótese alternativa (H_1): Existe diferença significativa entre o desempenho obtido por A com respeito ao desempenho de B.

O teste de “Wilcoxon Pareado” é então aplicado sobre a pontuação dos dois times, obtendo um valor-p de 0,04883, abaixo do nível de significância α de 5%, o que inva-

lida a hipótese nula, permitindo concluir (com menos de 5% de chance de estar errado) que TG4 é melhor que TG2.

O resultado do teste de “Wilcoxon Pareado” para todos os confrontos neste cenário é apresentado na Tabela 10. Através desses resultados é possível representar a ordem parcial dos times em razão do seu desempenho, como mostrado na Figura 23.

Tabela 10: Cenário SC3_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1							
TG2	0,005859		0,6953	0,04883	0,02734	0,7695	0,625
TG3	0,01953	0,6953		0,1602	0,03711	0,375	0,03711
TG4	0,001953	0,04883	0,1602		0,1602	0,08398	0,02734
TG6	0,003906	0,02734	0,03711	0,1602		0,1602	0,2324
TG8	0,03711	0,7695	0,375	0,08398	0,1602		0,625
TG10	0,01367	0,625	0,03711	0,02734	0,2324	0,625

Figura 23: Cenário SC3_SS - Grafo de ordem parcial dos times.

A ordem parcial é obtida desenhando uma aresta do time A para o B sempre que existe uma diferença significativa entre o desempenho dos dois times. A direção da aresta vai do time de maior desempenho para o de menor desempenho e são omitidas as arestas que podem ser extrapoladas através de um fecho transitivo, ou seja, aquelas arestas para as quais já existe um caminho ligando os dois times. As arestas tracejadas representam os confrontos para os quais o teste de “Wilcoxon Pareado” esteve perto de detectar uma diferença significativa entre dois times (valor-p menor que 0,15 e ao menos 7 vitórias para um dos times). Por fim, é possível que um mesmo time apareça duas ou mais vezes no grafo de ordem parcial, como mostrado na Figura 23, na qual TG3 aparece duas vezes. Isso é necessário para que possamos representar que TG3 obteve um desempenho melhor que TG10, mas o mesmo não foi observado em relação à TG6 e TG10.

Este tipo de análise foi feita em todos os confrontos e cenários, a fim de estabelecer

se com as 10 simulações é possível ou não concluir (com uma certa margem de erro) que um time é melhor que outro em determinado mapa. Os resultados obtidos em cada cenário são apresentados no Apêndice A.

5.3 Análise dos Resultados

Nesta seção, são apresentadas as análises dos resultados obtidos para os experimentos descritos na seção 5.1. Nesta análise, avaliou-se o impacto do tamanho, homogeneidade, dispersão e número de *clusters* nos times com diferentes números de pelotões.

5.3.1 Efeito do tamanho dos *clusters*

Para analisar o efeito do tamanho dos *clusters*, escolheu-se comparar os resultados obtidos pelos diversos times nos cenários SC1_P e SC1_G. Como mostra a Figura 24, tais cenários diferem em relação ao tamanho do *cluster*, sendo que SC1_G possui um *cluster* com valor maior e mais espalhado no grafo que SC1_P.

Figura 24: SC1_P vs SC1_G - Mapas.

O grafo de ordem parcial do desempenho dos times para estes dois cenários é apresentado na Figura 25. Comparando os resultados obtidos, é possível notar que:

1. TG10 tem desempenho melhor que TG1 apenas em SC1_G;
2. TG6 tem desempenho melhor que TG3 e TG4 em SC1_G, enquanto o contrário ocorre em SC1_P.

Figura 25: SC1_P vs SC1_G - Grafos de ordem parcial.

Em relação ao primeiro item, TG10 não consegue apresentar um desempenho melhor que TG1 em SC1_P. Como o *cluster* é pequeno, alguns de seus pelotões acabam ocupando áreas pequenas e de pouco valor fora do *cluster*. Além disso, o número de agentes de TG10 que disputam o *cluster* não é suficiente para vencer TG1 caso este ocupe primeiro a área de grande valor. Assim, o time que vence a partida é normalmente aquele que domina primeiro o *cluster*, não existindo assim diferença de desempenho entre os dois times.

Já em SC1_G, como o *cluster* é grande, este acaba sendo dividido em várias zonas. Assim, os pelotões de TG10 tendem a ocupar as zonas do *cluster* e não as áreas fora deste. Neste caso, TG10 consegue facilmente dominar uma área maior do *cluster* que TG1, como mostrado Figura 26, ganhando a maioria das partidas.

Figura 26: SC1_P vs SC1_G - TG1 (verde) vs TG10 (azul)

Este mesmo raciocínio se aplica ao segundo item. TG6 tem desempenho inferior à TG3 e TG4 em SC1_P, pois alguns de seus pelotões acabam ocupando áreas fora do *cluster*, já que este é dividido em um número menor de zonas. Ao contrário, em SC1_G os pelotões de TG6 tendem a ocupar todo ou a maior parte do *cluster*, conforme apresentado na Figura 27.

Figura 27: SC1_P vs SC1_G - TG6 (azul) vs TG3 (verde)

Nota-se também que os times que possuem um número de pelotões igual ou próximo ao número de zonas no *cluster* apresentam desempenho melhor que os outros times. Nos cenários aqui estudados, o *cluster* em SC1_P pode ser dividido em 2 zonas maiores e 2 menores, enquanto SC1_G possui até 6 zonas de grade valor, como mostrado na Figura 28. Assim, temos TG2, TG3 e TG4 com melhores desempenhos em SC1_P, e TG4 e TG6 com melhor desempenho em SC1_G.

Figura 28: SC1_P vs SC1_G - Divisão dos *clusters* em zonas.

Por fim, pode-se concluir que é melhor que o time possua um número de pelotões

igual ao número de zonas no *cluster* a ser conquistado. Um número menor de pelotões não é efetivo em conquistar e defender todo o *cluster*, enquanto um número maior acaba fazendo com que alguns pelotões se dispersem para fora do *cluster*, dificultando a sua conquista pelo time.

5.3.2 Efeito da homogeneidade dos *clusters*

Os cenários SC2_SS e SC2_NS diferem em relação à homogeneidade dos *clusters*, sendo que no primeiro os *clusters* possuem valor aproximadamente igual, enquanto no segundo um dos *clusters* é menor que o outro, como apresentado na Figura 29.

Figura 29: SC2_SS vs SC2_NS - Mapas.

O grafo de ordem parcial dos times para estes dois cenários é apresentado na Figura 30, onde é possível observar que TG1, TG2 e TG3 apresentam um desempenho melhor no cenário homogêneo.

Figura 30: SC2_SS vs SC2_NS - Grafos de ordem parcial.

Isto acontece pois, no cenário não homogêneo, estes times podem acabar designando todos ou um grande número de agentes a ocuparem o *cluster* menor, enquanto um time com mais pelotões tende a mandar um número menor de agentes, empregando assim a maioria dos seus agentes na conquista do *cluster* de maior valor. Tal fato é ilustrado na partida mostrada na Figura 31. Neste caso, TG8 consegue facilmente dominar o *cluster* de maior valor, pois utiliza um número maior de agentes para isto, ao passo que TG3 domina o outro cluster, mas sem pontuar tanto quanto TG8.

Figura 31: SC2_SS vs SC2_NS - TG8 (azul) vs TG3 (verde)

Além disso, como existem apenas dois *clusters* nestes cenários, pode acontecer de um time acabar não tomando conhecimento de um deles por um grande período de tempo, seja porque os exploradores demoram a chegar a um dos cantos do mapa, seja porque um time ocupando um dos clusters impede que os exploradores do oponente explorem aquela área. Neste caso, no cenário não homogêneo, isto pode prejudicar o time que inicialmente tenta conquistar apenas o *cluster* menor, pois ele irá pontuar menos que o oponente que ocupa o *cluster* maior. Um exemplo é mostrado na Figura 32, onde TG2 tenta conquistar apenas o cluster menor, enquanto TG6 domina o cluster maior. Neste caso, mesmo com uma área menor, TG6 pontua mais que TG2.

Dois outros cenários em que foi possível verificar a influência da homogeneidade dos *clusters* foram os cenários SC4_SN e SC4_NN. Como pode ser visto na Figura 33, no cenário SC4_NN existem dois *clusters* grandes e dois pequenos, enquanto que em SC4_SN os *clusters* são homogêneos.

Na Figura 34 é apresentado o grafo de ordem parcial dos desempenhos dos times para os cenários SC4_SN e SC4_NN.

Figura 32: SC2_SS vs SC2_NS - TG6 (azul) vs TG2 (verde)

Figura 33: SC4_SN vs SC4_NN - Mapas.

Figura 34: SC4_SN vs SC4_NN - Grafos de ordem parcial.

Comparando os resultados obtidos, observou-se que neste caso TG2 apresenta um desempenho melhor no cenário não homogêneo. Em SC4_NN, TG2 consegue equilibrar ou vencer as partidas quando os seus pelotões são designados a ocupar os dois maiores *clusters* no mapa, enquanto o oponente usa parte ou a totalidade dos seus

agentes para dominar os *clusters* menores. Um exemplo disto é apresentado na Figura 35, onde TG2 vence TG3.

Figura 35: SC4_SN vs SC4_NN - TG3 (azul) vs TG2 (verde)

Dados os resultados apresentados para os dois pares de cenários, conclui-se primeiramente que encontrar e ocupar os *clusters* de maior valor é fundamental em cenários não homogêneos. Em segundo lugar, a não homogeneidade dos *clusters* pode acabar prejudicando ou ajudando times com um número pequeno de pelotões. Ela prejudica quando o número de *clusters* no mapa é pequeno, pois times com um número maior de pelotões irão se concentrar em ocupar os *clusters* de maior valor, enquanto os times com menos pelotões acabam eventualmente usando um número muito grande de agentes para ocupar um *cluster* menos importante. Já em cenários com um número grande de *clusters*, a não homogeneidade pode ajudar os times com poucos pelotões, pois existe uma chance que estes venham a dominar os *clusters* de maior valor, enquanto os times com um número maior de pelotões acabam espalhando seus agentes em *clusters* menores.

5.3.3 Efeito da dispersão dos *clusters*

Para analisar o efeito da dispersão dos *clusters*, compararam-se os cenários SC2_SN e SC2_SS, que diferem apenas em relação à distância entre as duas áreas de grande valor (Figura 36).

O grafo de ordem parcial dos times nestes cenários é apresentado na Figura 37, onde é possível notar que TG3 e TG4 apresentam um desempenho melhor que TG2 no cenário SC2_SN com os *clusters* não dispersos ou mais próximos.

Figura 36: SC2_SN vs SC2_SS - Mapas.

Figura 37: SC2_SN vs SC2_SS - Grafos de ordem parcial.

Verificou-se que em SC2_SN a proximidade dos dois *clusters* facilita o ataque de um time sobre o outro, já que os agentes atacados precisam percorrer distâncias menores para serem reparados. Isto favorece tanto a infantaria quanto eventuais pelotões que venham a entrar na disputa por um *cluster* vizinho. Assim, quando um dos pelotões de TG3 ou TG4 tenta dominar o *cluster* ocupado por TG2, ele tem uma maior chance de sucesso, conseguindo muitas vezes encurralar TG2. Além disso, dada a proximidade, TG3 e TG4 acabam em geral dominando a área entre os *clusters*, como mostrado na Figura 38.

Os cenários SC6_SS e SC6_SN também diferem apenas em relação a dispersão dos *clusters*, conforme mostra a Figura 39. Neste caso, conforme os grafos de ordem parcial apresentados na Figura 40, TG8 obteve um desempenho melhor no cenário SC6_SN, onde os *clusters* estão mais próximos uns dos outros.

Figura 38: SC2_SN vs SC2_SS - TG3 (verde) vs TG2 (azul)

Figura 39: SC6_SS vs SC6_SN - Mapas.

Figura 40: SC6_SS vs SC6_SN - Grafos de ordem parcial.

Em SC6_SS, TG3 e TG4 conseguem dominar áreas grandes, que muitas vezes englobam dois *clusters* vizinhos, enquanto TG8 ocupa áreas menores e não consegue atacar eficientemente as áreas grandes do oponente. No entanto, em SC6_SN, com

a proximidade dos *clusters*, os pelotões de TG8 tendem a ficar mais próximos, facilitando o seu ataque ao oponente, e fazendo com que TG8 consiga as vezes dominar áreas que englobam mais de um *cluster* ao mesmo tempo, como mostrado na Figura 41.

Figura 41: SC6_SS vs SC6_SN - TG4 (verde) vs TG8 (azul)

Desta forma, em relação a dispersão é possível concluir que *clusters* menos dispersos facilitam o ataque, pois os agentes danificados são reparados rapidamente e, além disso, ajudam que times com um número maior de pelotões formem áreas maiores do que conseguiriam se os *clusters*, e consequentemente os agentes, estivessem dispersos.

5.3.4 Efeito do número de *clusters*

Finalmente, para analisar o efeito do número de *clusters*, escolheu-se comparar os resultados obtidos pelos diversos times nos cenários SC3_SS, SC4_SS e SC5_SS. Tais cenários encontram-se representados na Figura 42, e seus grafos de ordem parcial na Figura 43. Uma análise de desempenho dos times mostra as seguintes relações para os times TG3 e TG10:

- **SC3_SS:** TG3 tem desempenho melhor que TG10;
- **SC4_SS:** TG3 e TG10 tem desempenho equivalentes;
- **SC5_SS:** TG10 tem desempenho melhor que TG3.

Figura 42: SC3_SS vs SC4_SS vs SC5_SS - Mapas.

Figura 43: SC3_SS vs SC4_SS vs SC5_SS - Grafos de ordem parcial.

Em geral, para cenário homogêneos e dispersos como estes, nota-se que o ideal é que o número de pelotões seja igual ao número de *clusters*, ou de zonas no mapa. Um número menor de pelotões faz com que o time não cubra todos os *clusters*, podendo assim o oponente dominar facilmente estas áreas não cobertas. Um número maior de pelotões, por sua vez, pode fazer com que o time desperdice agentes alocando-os para áreas de pouco valor, enfraquecendo o poder de ataque e defesa dos pelotões.

5.4 Considerações Finais

Neste capítulo foram apresentados diversos resultados que mostraram a influência de características do ambiente (tamanho, homogeneidade, dispersão e número de *clusters*) sobre times com diferentes valores para um parâmetro organizacional (o número de pelotões). Em resumo, concluiu-se que:

1. *Clusters* maiores são bons para times com um número maior de pelotões;
2. Em cenários não homogêneos, é importante encontrar rapidamente os *clusters* de maior valor;
3. A não homogeneidade pode ser boa ou ruim para times com poucos pelotões, dependendo do número de *clusters* no mapa;
4. *Clusters* menos dispersos facilitam o ataque e são bons para times com um número maior de pelotões;
5. O ideal é que os times possuam tantos pelotões quanto *clusters* no mapa.

6 CONCLUSÕES

Determinar a organização mais adequada para um cenário específico é um problema-chave em SMA. Ao mesmo tempo que a organização tem como objetivo restringir o comportamento dos agentes, a fim de aumentar a eficiência global do sistema, esta pode afetar de forma adversa o desempenho do SMA. Assim, abordagens empíricas para a avaliação de organizações fornecem indícios valiosos sobre os custos e benefícios de diferentes configurações organizacionais, que podem ajudar na definição da estrutura e esquema a serem adotados.

Desta forma, este trabalho fez uma avaliação do impacto da mudança de um parâmetro organizacional no desempenho de um SMA. Em particular, estudou-se como alterações no número de grupos pode influenciar o desempenho de um time de agentes que competem no *Multi-Agent Programming Contest*. Além disto, o trabalho avaliou o impacto destas mudanças em ambientes com características distintas em relação ao tamanho, homogeneidade, dispersão e número de *clusters*.

Para validar as nossas observações, um teste estatístico, o teste de “Wilcoxon Paired”, foi utilizado para detectar diferenças no desempenho das organizações. Os resultados realizados permitiram concluir, por exemplo, que times com um número grande de grupos são mais eficientes em cenários homogêneos, não dispersos e com um número igual de *clusters*.

Além disso, os resultados mostram que não existe uma organização melhor que as outras em todos os cenários estudados. Assim, dado que os times não tem conhecimento a priori do cenário, o ideal seria que eles fossem capazes de se reorganizar de acordo com as características do ambiente e do adversário. Como discutido em (HÜBNER, 2003), a reorganização é um aspecto importante em SMA, uma vez que o ambiente na sua maioria das vezes não é estático. Assim, SMA devem ser capazes de alterar sua organização a fim de se adaptar a mudanças no ambiente.

Deste modo, uma possibilidade de trabalho futuro é utilizar os resultados obtidos neste trabalho para o desenvolvimento de um time capaz de se reorganizar dinamicamente. Para isto, o time poderia explorar inicialmente o mapa por algum tempo para tentar identificar as características do ambiente, e então definir a melhor organização a ser utilizada de acordo com os resultados que obtivemos. Depois de um certo intervalo de tempo, o desempenho do time poderia ser reavaliado, e em caso de um resultado negativo, novamente um ciclo de exploração poderia ser realizado.

Uma outra possibilidade de trabalho futuro é aumentar o número de diferentes cenários testados, e também avaliar o impacto de outros parâmetros organizacionais, mudando-se, por exemplo, o número de agentes encarregados de atacar os adversários, o número de agentes exploradores, etc. Tais parâmetros, inclusive, também poderiam ser objeto de uma alteração dinâmica: por exemplo, caso uma área de grande valor seja atacada, mais agentes poderiam alterar dinamicamente seu papel com o intuito de defender tal área.

Por fim, é importante salientar que, no decorrer deste trabalho, foram publicados quatro artigos sobre o “LTI-USP Team”: (FRANCO; ROSSET; SICHMAN, 2013), (FRANCO; SICHMAN, 2013), (FRANCO; SICHMAN, 2013), (FRANCO; SICHMAN, 2014).

REFERÊNCIAS

ARGENTE, E.; JULIAN, V.; BOTTI, V. Multi-Agent System Development Based on Organizations. *Electronic Notes in Theoretical Computer Science*, v. 150, n. 3, p. 55–71, 2006. ISSN 15710661. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S157106610600329X>.

BEHRENS, T.; KÖSTER, M.; SCHLESINGER, F. The multi-agent programming contest 2011: a résumé. *Programming Multi-Agent Systems*, p. 155–172, 2012. Disponível em: <http://www.springerlink.com/index/C81P265085N04754.pdf>.

BEHRENS, T. M.; DIX, J.; HINDRIKS, K. V. The Environment Interface Standard for Agent-Oriented Programming - Platform Integration Guide and Interface Implementation Guide. *Department of Informatics, Clausthal University of Technology, Technical Report*, IfI-09-10, 2009.

BOISSIER, O. et al. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming*, Elsevier B.V., out. 2011. ISSN 01676423. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S016764231100181X>.

BOISSIER, O.; HÜBNER, J.; SICHMAN, J. Organization oriented programming: From closed to open organizations. In: *Proceedings of the 7th international conference on Engineering societies in the agents world VII - ESAW'06*. Dublin: Springer-Verlag, 2007. p. 86 – 105. Disponível em: <http://www.springerlink.com/index/87811466T2238742.pdf>.

BORDINI, R.; HÜBNER, J. *Jason: A Java-based interpreter for an extended version of AgentSpeak*. Mar, 2007. Disponível em: <http://jason.sourceforge.net/Jason.pdf>.

BORDINI, R.; HÜBNER, J.; WOOLDRIDGE, M. *Programming multi-agent systems in AgentSpeak using Jason*. Wiley-Blackwell, 2007. ISBN 9780470057476. Disponível em: <http://jason.sourceforge.net/jBook/jBook/Home.html>.

BRATMAN, M. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987. ISBN 9780674458185. Disponível em: <http://www.press.uchicago.edu/ucp/books/book/distributed/I/bo3629095.html>.

CARLEY, K. M. Computational and mathematical organization theory: Perspective and directions. *Computational and Mathematical Organization Theory*, Kluwer Academic Publishers, v. 1, n. 1, p. 39–56, 1995. ISSN 1381-298X. Disponível em: <http://dx.doi.org/10.1007/BF01307827>.

CARLEY, K. M.; GASSER, L. Multiagent systems. In: WEISS, G. (Ed.). Cambridge, MA, USA: MIT Press, 1999. cap. Computational Organization Theory, p. 299–330. ISBN 0-262-23203-0. Disponível em: <http://dl.acm.org/citation.cfm?id=305606.305613>.

DEMAZEAU, Y. From interactions to collective behaviour in agent-based systems. In: *Proceedings of the 1st. European Conference on Cognitive Science*. Saint-Malo: [s.n.], 1995. p. 117–132.

DEMAZEAU, Y.; MÜLLER, J. *Decentralized Artificial Intelligence*. [S.l.]: Elsevier, 1990. ISBN 0444887059, 9780444887054.

DIGNUM, V. *A model for organizational interaction: based on agents, founded in logic*. Tese (Doutorado) — Universiteit Utrecht, 2004.

ESTEVA, M.; CRUZ, D.; SIERRA, C. ISLANDER: an electronic institutions editor. In: *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2002. (AAMAS '02), p. 1045–1052. ISBN 1-58113-480-0. Disponível em: <http://doi.acm.org/10.1145/545056.545069>.

ESTEVA, M. et al. AMELI: An Agent-Based Middleware for Electronic Institutions. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2004. (AAMAS '04), p. 236–243. ISBN 1-58113-864-4. Disponível em: <http://dx.doi.org/10.1109/AAMAS.2004.56>.

FERBER, J.; GUTKNECHT, O.; MICHEL, F. From agents to organizations: an organizational view of multi-agent systems. *Agent-Oriented Software Engineering IV*, n. July 2003, p. 214–230, 2004. Disponível em: <http://www.springerlink.com/index/pkb4mtc84cr7q77p.pdf>.

FININ, T. et al. *Specification of the KQML Agent-Communication Language*. 1994.

FRANCO, M. R.; ROSSET, L. M.; SICHMAN, J. S. LTI-USP Team: A JaCaMo Based MAS for the MAPC 2012. In: DASTANI, M.; HÜBNER, J. F.; LOGAN, B. (Ed.). *Programming Multi-Agent Systems*. Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7837). p. 224–233. ISBN 978-3-642-38699-2. Disponível em: http://dx.doi.org/10.1007/978-3-642-38700-5_15.

FRANCO, M. R.; SICHMAN, J. S. Improving the LTI-USP Team: A New JaCaMo Based MAS for the MAPC 2013. In: COSSENTINO, M.; SEGHTROUCHNI, A. F.; WINIKOFF, M. (Ed.). *Engineering Multi-Agent Systems*. Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 8245). p. 339–348. ISBN 978-3-642-45342-7. Disponível em: http://dx.doi.org/10.1007/978-3-642-45343-4_19.

FRANCO, M. R.; SICHMAN, J. S. Using the JaCaMo framework to develop a MAS for the MAPC 2012 "Agents on Mars" scenario. In: BRANDÃO, A. A. F.; BORDINI, R. H.; SICHMAN, J. S. (Ed.). *Anais do VII Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações - WESAAC*. São Paulo: [s.n.], 2013. p. 91–96.

FRANCO, M. R.; SICHMAN, J. S. Comparing and Evaluating Organizational Models: A Multi-Agent Programming Contest Case Study. In: OREN, N.; TELANG, P. (Ed.). *The 17th International Workshop on Coordination, Organisations, Institutions and Norms - COIN 2014*. Paris, France: [s.n.], 2014.

HARRISON, J. R. et al. Simulation modeling in organizational and management research. *Academy of Management Review*, Academy of Management, v. 32, n. 4, p. 1229–1245, 2007.

HORLING, B.; LESSER, V. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 2004. Disponível em: <http://dl.acm.org/citation.cfm?id=1103864>.

HORLING, B.; LESSER, V. Analyzing, Modeling and Predicting Organizational Effects in a Distributed Sensor Network. *Journal of the Brazilian Computer Society, Special Issue on Agents Organizations*, v. 11, n. 1, p. 9–30, July 2005. Disponível em: <http://mas.cs.umass.edu/paper/379>.

HOWDEN, N. et al. JACK Intelligent Agents - Summary of an Agent Infrastructure. In: *5th International conference on autonomous agents*. [s.n.], 2001. Disponível em: http://users.cs.cf.ac.uk/O.F.Rana/agents2001/papers/18_howden.pdf.

HÜBNER, J.; SICHMAN, J.; BOISSIER, O. A model for the structural, functional, and deontic specification of organizations in multiagent systems. *Advances in Artificial Intelligence*, n. 2, 2002. Disponível em: <http://www.springerlink.com/index/84uevb0c6b6436wr.pdf>.

HÜBNER, J. F. *Um Modelo de Reorganização de Sistemas Multiagentes*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 2003.

HÜBNER, J. F. et al. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems*, v. 20, n. 3, p. 369–400, abr. 2009. ISSN 1387-2532. Disponível em: <http://www.springerlink.com/index/10.1007/s10458-009-9084-y>.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. S-moise+: A middleware for developing organised multi-agent systems. *Lecture Notes in Computer Science*, v. 3913, p. 64–77, 2006.

HÜBNER, J. F.; SICHMAN, J. S.; BOISSIER, O. Developing Organised Multiagent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels. *International Journal of Agent-Oriented Software Engineering*, Inderscience Publishers, Geneva, v. 1, n. 3/4, p. 370–395, dez. 2007. ISSN 1746-1375.

JENNINGS, N. R.; WOOLDRIDGE, M. Agent-Oriented Software Engineering. In: *Handbook of Agent Technology*. [S.l.]: AAAI/MIT Press, 2001.

JENSEN, A. S. *Multi-Agent Systems : An Investigation of the Advantages of Making Organizations Explicit*. Tese (Mestrado) — Technical University of Denmark, 2010.

JOHNSON, R. A. *Miller and Freund's Probability and Statistics for Engineers*. 6. ed. [S.l.]: Prentice Hall, 2000. ISBN 0130141585.

KITIO, R. et al. Organisational artifacts and agents for open multi-agent organisations: giving the power back to the agents. In: *Proceedings of the 2007 international conference on Coordination, organizations, institutions, and norms in agent systems*

III - COIN'07. Honolulu: Springer-Verlag, 2008. p. 171 – 186. Disponível em: <http://www.springerlink.com/index/h4044303q272j3mm.pdf>.

LEMAÎTRE, C.; EXCELENTE, C. Multi-Agent Organization Approach. In *Proceedings of the second Iberoamerican Workshop on Distributed Artificial Intelligence and Multi-Agent systems*, p. 7–16, 1998.

MORLEY, D.; MYERS, K. The SPARK Agent Framework. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*. Washington, DC, USA: IEEE Computer Society, 2004. (AAMAS '04), p. 714–721. ISBN 1-58113-864-4. Disponível em: <http://dx.doi.org/10.1109/AAMAS.2004.267>.

NIKOS, V. *A Concise Introduction to Multiagent Systems and Distributed Artificial Intelligence*. 1st. ed. Morgan and Claypool Publishers, 2007. ISBN 1598295268. Disponível em: <http://dl.acm.org/citation.cfm?id=1534862>.

OKOUYA, D.; DIGNUM, V. Operetta: a prototype tool for the design, analysis and development of multi-agent organizations. In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: demo papers*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2008. (AAMAS '08), p. 1677–1678. Disponível em: <http://dl.acm.org/citation.cfm?id=1402744.1402761>.

OMICINI, A.; RICCI, A.; VIROLI, M. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, v. 17, n. 3, p. 432–456, maio 2008. ISSN 1387-2532. Disponível em: <http://www.springerlink.com/index/10.1007/s10458-008-9053-x>.

POKAHR, A.; BRAUBACH, L.; LAMERSDORF, W. Jadex: A BDI Reasoning Engine. In: BORDINI, R. et al. (Ed.). *Multi-Agent Programming*. Springer US, 2005, (Multiagent Systems, Artificial Societies, and Simulated Organizations, v. 15). p. 149–174. ISBN 978-0-387-24568-3. Disponível em: http://dx.doi.org/10.1007/0-387-26350-0_6.

PRIETULA, M. J.; CARLEY, K. M.; GASSER, L. (Ed.). *Simulating Organizations: Computational Models of Institutions and Groups*. Cambridge, MA, USA: MIT Press, 1998. ISBN 0-262-66108-X.

RAO, A. AgentSpeak(L): BDI agents speak out in a logical computable language. In: VELDE, W.; PERRAM, J. (Ed.). *Agents Breaking Away*. Springer Berlin Heidelberg, 1996, (Lecture Notes in Computer Science, v. 1038). p. 42–55. ISBN 978-3-540-60852-3. Disponível em: <http://dx.doi.org/10.1007/BFb0031845>.

RICCI, A.; PIUNTI, M.; VIROLI, M. Environment programming in multi-agent systems: an artifact-based perspective. *Autonomous Agents and Multi-Agent Systems*, v. 23, n. 2, p. 158–192, jun. 2010. ISSN 1387-2532. Disponível em: <http://www.springerlink.com/index/10.1007/s10458-010-9140-7>.

RICCI, A.; VIROLI, M.; OMICINI, A. CArtaGO: An infrastructure for engineering computational environments in MAS. *3rd Inter. Workshop "Environments for Multi-Agent Systems"(E4MAS)*, 2006. Disponível em: <http://lia.deis.unibo.it/~ao/pubs/pdf/2006/e4mas-rvo.pdf>.

RICCI, A.; VIROLI, M.; OMICINI, A. CArtaGO: A Framework for Prototyping Artifact-Based Environments in MAS. *Environments for Multi-Agent Systems III*, p. 67–86, 2007. Disponível em: <http://www.springerlink.com/index/D1047836893J14Q6.pdf>.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010. (Prentice Hall Series in Artificial Intelligence). ISBN 9780136042594. Disponível em: <http://aima.cs.berkeley.edu/>.

SICHTMAN, J.; DIGNUM, V.; CASTELFRANCHI, C. Agents' organizations: a concise overview. *Journal of the Brazilian Computer Society*, v. 11, p. 3 – 8, dez. 2005. ISSN 1873-5126.

SYCARA, K. Multiagent systems. *AI magazine*, v. 19, n. 2, p. 79–92, 1998. Disponível em: <http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1370>.

TRIOLA, M. *Introdução á estatística*. [S.l.]: LTC, 2008. ISBN 9788521615866.

WEISS, G. *Multiagent systems: a modern approach to distributed artificial intelligence*. Cambridge, MA, USA: MIT Press, 1999. ISBN 0262232030.

WOOLDRIDGE, M. *An introduction to multiagent systems*. Cambridge: Cambridge University Press, 2002. ISBN 9780511811654.

APÊNDICE A - RESULTADOS DOS EXPERIMENTOS

A.1 Cenário SC1_P

Figura 44: Cenário SC1_P - Mapa.

Figura 45: Cenário SC1_P - Grafo de ordem parcial.

Tabela 11: Cenário SC1_P - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2	TG1 (0) vs (10) TG3	TG1 (0) vs (10) TG4
32072,9 vs 44960,7 ± 7461,6 ± 3217,8	33069,4 vs 44996 ± 3578,8 ± 3483,1	31628,1 vs 43702,1 ± 5064,1 ± 2862,3
TG1 (1) vs (9) TG6	TG1 (3) vs (7) TG8	TG1 (5) vs (5) TG10
34461,4 vs 41530,1 ± 3659,6 ± 2781,2	35905 vs 40283,5 ± 4916,8 ± 4452,5	35767,4 vs 35960,7 ± 4857,7 ± 4754,0
TG2 (3) vs (7) TG3	TG2 (3) vs (7) TG4	TG2 (6) vs (4) TG6
35849,7 vs 39357,7 ± 3494,3 ± 5556,7	34872,8 vs 38132,9 ± 4411,0 ± 4156,6	37310,3 vs 36711,1 ± 4218,7 ± 4277,0
TG2 (8) vs (2) TG8	TG2 (10) vs (0) TG10	TG3 (1) vs (9) TG4
39775,7 vs 31288 ± 3802,7 ± 4210,1	40610 vs 33157,8 ± 3996,7 ± 4250,0	32955,2 vs 36611 ± 2866,0 ± 3875,6
TG3 (10) vs (0) TG6	TG3 (9) vs (1) TG8	TG3 (10) vs (0) TG10
41010,5 vs 30858,3 ± 5977,2 ± 3396,4	39763 vs 31691,4 ± 3039,9 ± 3028,9	45308 vs 27554,6 ± 3924,2 ± 3385,9
TG4 (10) vs (0) TG6	TG4 (10) vs (0) TG8	TG4 (10) vs (0) TG10
39581,5 vs 30620,5 ± 4424,8 ± 2726,8	40632,2 vs 31265,5 ± 5356,9 ± 4982,4	44151,5 vs 28234,5 ± 2670,2 ± 2653,0
TG6 (8) vs (2) TG8	TG6 (8) vs (2) TG10	TG8 (7) vs (3) TG10
37939,9 vs 33122,3 ± 4609,9 ± 4727,1	40081,4 vs 29366,8 ± 4036,0 ± 4813,9	36185 vs 30160,7 ± 5884,9 ± 4110,8

Tabela 12: Cenário SC1_P - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,001953	0,001953	0,005859	0,1309	1
TG2	0,001953		0,2754	0,2754	0,7695	0,03711	0,001953
TG3	0,001953	0,2754		0,04883	0,001953	0,003906	0,001953
TG4	0,001953	0,2754	0,04883		0,001953	0,009766	0,001953
TG6	0,005859	0,7695	0,001953	0,001953		0,1309	0,01367
TG8	0,1309	0,03711	0,003906	0,009766	0,1309		0,08398
TG10	1	0,001953	0,001953	0,001953	0,01367	0,08398	

A.2 Cenário SC1_G

Figura 46: Cenário SC1_G - Mapa.

Tabela 13: Cenário SC1_G - Média e desvio padrão dos confrontos.

TG1 (1) vs (9) TG2	TG1 (1) vs (9) TG3	TG1 (0) vs (10) TG4
55876,5 vs 61057,4 ± 7177,5 ± 8762,6	50880,3 vs 61267,2 ± 10403,6 ± 7402,3	52886,2 vs 65073,8 ± 6898,4 ± 3633,0
TG1 (0) vs (10) TG6	TG1 (1) vs (9) TG8	TG1 (1) vs (9) TG10
47964,6 vs 64818,6 ± 9352,5 ± 3261,0	47406,2 vs 61251,7 ± 9925,4 ± 3641,2	50368,9 vs 57007,3 ± 10056,5 ± 5721,3
TG2 (3) vs (7) TG3	TG2 (1) vs (9) TG4	TG2 (0) vs (10) TG6
55639,5 vs 58724,8 ± 6448,4 ± 5231,5	54187,2 vs 61163,2 ± 6965,8 ± 4853,9	54641,7 vs 61946,5 ± 5587,6 ± 4526,3
TG2 (1) vs (9) TG8	TG2 (5) vs (5) TG10	TG3 (2) vs (8) TG4
47173,6 vs 58412,6 ± 5891,5 ± 6565,8	53635,7 vs 52532,1 ± 5453,2 ± 6369,3	57890,1 vs 61096,1 ± 6628,8 ± 8496,8
TG3 (2) vs (8) TG6	TG3 (4) vs (6) TG8	TG3 (9) vs (1) TG10
56822 vs 59803,7 ± 8410,9 ± 5745,3	56521,6 vs 57722,1 ± 6323,3 ± 8549,6	55255,8 vs 50399,6 ± 6317,4 ± 6316,1
TG4 (3) vs (7) TG6	TG4 (8) vs (2) TG8	TG4 (7) vs (3) TG10
54139 vs 57689 ± 7610,1 ± 6578,6	56170 vs 54473,6 ± 6431,2 ± 7148,6	57565,9 vs 53401,6 ± 3541,3 ± 7698,1
TG6 (5) vs (5) TG8	TG6 (7) vs (3) TG10	TG8 (7) vs (3) TG10
50930,5 vs 50675,1 ± 3304,7 ± 5189,1	54072,7 vs 50761,6 ± 6299,6 ± 7498,1	52571,5 vs 48947,3 ± 3759,8 ± 6662,4

Tabela 14: Cenário SC1_G - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,005859	0,003906	0,001953	0,001953	0,003906	0,009766
TG2	0,005859		0,1309	0,005859	0,001953	0,003906	0,9219
TG3	0,003906	0,1309		0,01953	0,1055	0,625	0,02734
TG4	0,001953	0,005859	0,01953		0,04883	0,2754	0,1055
TG6	0,001953	0,001953	0,1055	0,04883		0,8457	0,1309
TG8	0,003906	0,003906	0,625	0,2754	0,8457		0,02734
TG10	0,009766	0,9219	0,02734	0,1055	0,1309	0,02734

Figura 47: Cenário SC1_G - Grafo de ordem parcial.

A.3 Cenário SC2_SN

Figura 48: Cenário SC2_SN - Mapa.

Tabela 15: Cenário SC2_SN - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2	TG1 (2) vs (8) TG3	TG1 (0) vs (10) TG4
56940,7 vs 74153,3 ± 12411,0 ± 9617,9	64279,4 vs 75418,7 ± 13803,6 ± 8482,5	63848,9 vs 73962,6 ± 9679,8 ± 10961,3
TG1 (0) vs (10) TG6	TG1 (4) vs (6) TG8	TG1 (5) vs (5) TG10
69588,6 vs 77518,4 ± 11348,7 ± 10656,6	67795,3 vs 75179,4 ± 12752,1 ± 13619,0	69178 vs 66770,7 ± 8275,6 ± 13225,4
TG2 (1) vs (9) TG3	TG2 (2) vs (8) TG4	TG2 (6) vs (4) TG6
73958,9 vs 80115,1 ± 8107,9 ± 7901,2	69356,9 vs 75490,9 ± 11166,2 ± 10245,0	70432,3 vs 68987,5 ± 13399,9 ± 11970,7
TG2 (3) vs (7) TG8	TG2 (6) vs (4) TG10	TG3 (3) vs (7) TG4
73773,1 vs 76707,1 ± 6882,8 ± 11860,7	67776,5 vs 67039,6 ± 9335,9 ± 10073,6	73915,8 vs 76862,3 ± 12468,5 ± 9950,7
TG3 (5) vs (5) TG6	TG3 (8) vs (2) TG8	TG3 (9) vs (1) TG10
76684,2 vs 73777,7 ± 8603,0 ± 10367,9	82364,9 vs 76637,3 ± 5285,3 ± 8753,9	77304,9 vs 56217,5 ± 8193,2 ± 19484,7
TG4 (5) vs (5) TG6	TG4 (8) vs (2) TG8	TG4 (9) vs (1) TG10
77881,3 vs 80312,7 ± 11459,6 ± 8118,0	70821,1 vs 59892,4 ± 9888,3 ± 13861,3	74289,7 vs 57218 ± 8474,5 ± 14718,5
TG6 (8) vs (2) TG8	TG6 (8) vs (2) TG10	TG8 (9) vs (1) TG10
69586,4 vs 59952,7 ± 13662,4 ± 16095,9	72661,7 vs 60749 ± 13446,8 ± 15078,5	71450,9 vs 55965,1 ± 14737,8 ± 14568,5

Tabela 16: Cenário SC2_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,01367	0,001953	0,001953	0,1602	0,6953
TG2	0,001953		0,003906	0,08398	0,8457	0,4316	0,9219
TG3	0,01367	0,003906		0,2324	0,4316	0,08398	0,005859
TG4	0,001953	0,08398	0,2324		0,4316	0,06445	0,005859
TG6	0,001953	0,8457	0,4316	0,4316		0,1055	0,009766
TG8	0,1602	0,4316	0,08398	0,06445	0,1055		0,01953
TG10	0,6953	0,9219	0,005859	0,005859	0,009766	0,01953

Figura 49: Cenário SC2_SN - Grafo de ordem parcial.

A.4 Cenário SC2_SS

Figura 50: Cenário SC2_SS - Mapa.

Tabela 17: Cenário SC2_SS - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2	TG1 (2) vs (8) TG3	TG1 (2) vs (8) TG4
65706,3 vs 77524,7 ± 8451,9 vs ± 8223,6	64045,6 vs 73052,4 ± 12420,2 vs ± 12636,1	70836,9 vs 77656,1 ± 13504,9 vs ± 11234,2
TG1 (4) vs (6) TG6	TG1 (4) vs (6) TG8	TG1 (5) vs (5) TG10
74277,6 vs 74858,1 ± 4340,6 vs ± 12637,6	66492,2 vs 72670,4 ± 15969,2 vs ± 10714,5	68025,7 vs 70822,7 ± 9725,5 vs ± 17335,6
TG2 (3) vs (7) TG3	TG2 (5) vs (5) TG4	TG2 (4) vs (6) TG6
74519,2 vs 70924,4 ± 6045,1 vs ± 22799,7	79502,4 vs 75951,5 ± 3697,0 vs ± 9724,3	72721,7 vs 72203,9 ± 11714,5 vs ± 12964,1
TG2 (7) vs (3) TG8	TG2 (5) vs (5) TG10	TG3 (4) vs (6) TG4
72989,2 vs 69840,2 ± 12787,7 vs ± 10357,8	70017 vs 69888,1 ± 10616,2 vs ± 13627,0	77993,8 vs 80213,9 ± 8871,0 vs ± 5649,8
TG3 (4) vs (6) TG6	TG3 (5) vs (5) TG8	TG3 (5) vs (5) TG10
70917,9 vs 76380,5 ± 15643,1 vs ± 11307,1	80399,3 vs 79727,4 ± 11223,1 vs ± 11271,3	72037,1 vs 67728,5 ± 7921,0 vs ± 20400,7
TG4 (5) vs (5) TG6	TG4 (8) vs (2) TG8	TG4 (9) vs (1) TG10
76742 vs 67746 ± 10657,4 vs ± 16123,7	73768,4 vs 64050,6 ± 11876,8 vs ± 13644,3	80058,4 vs 68178,5 ± 7389,8 vs ± 15676,7
TG6 (9) vs (1) TG8	TG6 (9) vs (1) TG10	TG8 (10) vs (0) TG10
72012,4 vs 60645,9 ± 10749,0 vs ± 10698,1	78318,1 vs 65953,1 ± 11796,7 vs ± 17934,3	70261,3 vs 60737,6 ± 13348,1 vs ± 14160,6

Tabela 18: Cenário SC2_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,06445	0,06445	0,8457	0,3223	0,6953
TG2	0,001953		0,625	0,4316	1	0,375	0,8457
TG3	0,06445	0,625		0,4316	0,4922	1	0,625
TG4	0,06445	0,4316	0,4316		0,03711	0,01953	0,009766
TG6	0,8457	1	0,4922	0,03711		0,04883	0,01367
TG8	0,3223	0,375	1	0,01953	0,04883		0,001953
TG10	0,6953	0,8457	0,625	0,009766	0,01367	0,001953

Figura 51: Cenário SC2_SS - Grafo de ordem parcial.

A.5 Cenário SC2_NS

Figura 52: Cenário SC2_NS - Mapa.

Tabela 19: Cenário SC2_NS - Média e desvio padrão dos confrontos.

TG1 (8) vs (2) TG2	TG1 (7) vs (3) TG3	TG1 (1) vs (9) TG4
64769,7 vs 63347,5 ± 16019,2 ± 5625,7	63911 vs 57823 ± 14411,0 ± 12316,3	56400,2 vs 74830,4 ± 7879,3 ± 8387,1
TG1 (0) vs (10) TG6	TG1 (2) vs (8) TG8	TG1 (1) vs (9) TG10
56448,1 vs 75905,2 ± 9937,7 ± 9866,9	57682,4 vs 69456,9 ± 9135,0 ± 17592,3	46372,3 vs 61480,3 ± 10424,5 ± 13490,0
TG2 (10) vs (0) TG3	TG2 (0) vs (10) TG4	TG2 (2) vs (8) TG6
78704,6 vs 67670,3 ± 2298,7 ± 4800,6	62220,6 vs 77864,8 ± 10953,6 ± 10350,6	60455,7 vs 72683,8 ± 10445,4 ± 13801,3
TG2 (4) vs (6) TG8	TG2 (3) vs (7) TG10	TG3 (0) vs (10) TG4
65395,7 vs 69888,6 ± 5457,3 ± 17445,4	62906 vs 66525 ± 5120,0 ± 15968,7	64774,8 vs 82508,9 ± 6085,8 ± 6344,2
TG3 (2) vs (8) TG6	TG3 (0) vs (10) TG8	TG3 (4) vs (6) TG10
64889,7 vs 72569,2 ± 10305,1 ± 11752,1	69921,4 vs 80323,4 ± 4247,5 ± 7946,7	59786,6 vs 64783,8 ± 8478,0 ± 19795,7
TG4 (9) vs (1) TG6	TG4 (10) vs (0) TG8	TG4 (10) vs (0) TG10
73642,9 vs 54638,4 ± 15072,9 ± 16026,7	71380,3 vs 46363,3 ± 9143,1 ± 13175,1	74479,1 vs 45107,8 ± 8751,5 ± 12896,6
TG6 (10) vs (0) TG8	TG6 (10) vs (0) TG10	TG8 (10) vs (0) TG10
72515,8 vs 47933,6 ± 12906,9 ± 14738,4	74213,4 vs 46514,4 ± 8081,8 ± 13207,3	68337,4 vs 43614,8 ± 8541,2 ± 11163,7

Tabela 20: Cenário SC2_NS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,1934	0,3223	0,003906	0,001953	0,04883	0,01953
TG2	0,1934		0,001953	0,001953	0,009766	0,3223	0,6953
TG3	0,3223	0,001953		0,001953	0,04883	0,001953	0,4316
TG4	0,003906	0,001953	0,001953		0,003906	0,001953	0,001953
TG6	0,001953	0,009766	0,04883	0,003906		0,001953	0,001953
TG8	0,04883	0,3223	0,001953	0,001953	0,001953		0,001953
TG10	0,01953	0,6953	0,4316	0,001953	0,001953	0,001953

Figura 53: Cenário SC2_NS - Grafo de ordem parcial.

A.6 Cenário SC3_SS

Figura 54: Cenário SC3_SS - Mapa.

Tabela 21: Cenário SC3_SS - Média e desvio padrão dos confrontos.

TG1 (1) vs (9) TG2	TG1 (2) vs (8) TG3	TG1 (0) vs (10) TG4
39796,8 vs 60742,8 ± 6381,7 vs ± 11095,5	58534,2 vs 74315,3 ± 15376,4 vs ± 10281,4	54058,2 vs 78117,2 ± 11453,4 vs ± 11100,3
TG1 (1) vs (9) TG6	TG1 (2) vs (8) TG8	TG1 (1) vs (9) TG10
60659,1 vs 81544,9 ± 10740,8 vs ± 9394,1	63224,6 vs 72080 ± 12654,5 vs ± 6383,1	65849,8 vs 78086,2 ± 11745,6 vs ± 13136,4
TG2 (4) vs (6) TG3	TG2 (2) vs (8) TG4	TG2 (2) vs (8) TG6
62978,7 vs 64721,9 ± 9038,2 vs ± 11823,1	56558,2 vs 66889,3 ± 11532,9 vs ± 6408,1	68547,6 vs 77210,4 ± 8314,6 vs ± 10080
TG2 (6) vs (4) TG8	TG2 (6) vs (4) TG10	TG3 (4) vs (6) TG4
70132,5 vs 69764,2 ± 10284,1 vs ± 9929,2	68331,7 vs 67067,5 ± 9248,2 vs ± 4713,3	66751,9 vs 74329,2 ± 8767,3 vs ± 16110,1
TG3 (1) vs (9) TG6	TG3 (6) vs (4) TG8	TG3 (8) vs (2) TG10
67016,5 vs 76053,6 ± 10309,7 vs ± 10474,9	76270,6 vs 72497,2 ± 10507,4 vs ± 7900,6	77370,6 vs 68079 ± 7555,7 vs ± 12098,6
TG4 (7) vs (3) TG6	TG4 (7) vs (3) TG8	TG4 (9) vs (1) TG10
75776,8 vs 69061,8 ± 8905,3 vs ± 7526,6	80947,5 vs 71958,7 ± 13542,3 vs ± 11523,3	80014 vs 67593,4 ± 9331 vs ± 8365,7
TG6 (3) vs (7) TG8	TG6 (5) vs (5) TG10	TG8 (6) vs (4) TG10
75528,8 vs 80511,7 ± 7501,4 vs ± 7076,2	77810 vs 70745,5 ± 8879,9 vs ± 8069,5	74553,9 vs 72141,3 ± 9521,1 vs ± 12960,5

Tabela 22: Cenário SC3_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,005859	0,01953	0,001953	0,003906	0,03711	0,01367
TG2	0,005859		0,6953	0,04883	0,02734	0,7695	0,625
TG3	0,01953	0,6953		0,1602	0,03711	0,375	0,03711
TG4	0,001953	0,04883	0,1602		0,1055	0,08398	0,02734
TG6	0,003906	0,02734	0,03711	0,1055		0,1602	0,2324
TG8	0,03711	0,7695	0,375	0,08398	0,1602		0,625
TG10	0,01367	0,625	0,03711	0,02734	0,2324	0,625

Figura 55: Cenário SC3_SS - Grafo de ordem parcial.

A.7 Cenário SC3_SN

Figura 56: Cenário SC3_SN - Mapa.

Tabela 23: Cenário SC3_SN - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2	TG1 (0) vs (10) TG3	TG1 (1) vs (9) TG4
64700,6 vs 75136,6 ± 10405,2 ± 10970,8	65393,3 vs 80267,4 ± 12635,5 ± 13790,3	64994,5 vs 75600,6 ± 9719,4 ± 10025,2
TG1 (1) vs (9) TG6	TG1 (0) vs (10) TG8	TG1 (1) vs (9) TG10
68039,5 vs 80952,5 ± 10976,6 ± 4511,2	63469,2 vs 83342,1 ± 11940,7 ± 9060,5	64301,7 vs 72211,9 ± 12386,6 ± 11279,5
TG2 (6) vs (4) TG3	TG2 (1) vs (9) TG4	TG2 (2) vs (8) TG6
77804,1 vs 77545,7 ± 9557,6 ± 11282,9	76410,7 vs 79936,7 ± 12465,9 ± 12057,4	68480,3 vs 73423,8 ± 9471,9 ± 10895,1
TG2 (2) vs (8) TG8	TG2 (9) vs (1) TG10	TG3 (3) vs (7) TG4
73362,8 vs 78474,2 ± 10036,3 ± 12430,7	77963,6 vs 70449,1 ± 7836,7 ± 6767,8	79446,9 vs 81250,7 ± 8327,1 ± 7636,5
TG3 (5) vs (5) TG6	TG3 (5) vs (5) TG8	TG3 (6) vs (4) TG10
72879,1 vs 72806,6 ± 9428,2 ± 10862,1	73298 vs 71927,7 ± 5486,2 ± 7729,3	75358,8 vs 71476,5 ± 12261,2 ± 7085,0
TG4 (5) vs (5) TG6	TG4 (5) vs (5) TG8	TG4 (6) vs (4) TG10
74636,7 vs 76616,9 ± 12680,1 ± 5679,1	70797,9 vs 70766,9 ± 11162,0 ± 10942,7	75577,1 vs 71635,4 ± 10418,1 ± 9938,5
TG6 (5) vs (5) TG8	TG6 (6) vs (4) TG10	TG8 (7) vs (3) TG10
77268,2 vs 75299,4 ± 7654,1 ± 9480,4	72335,4 vs 71183,8 ± 8555,2 ± 8440,5	70374,2 vs 66349,1 ± 8407,8 ± 7580,4

Tabela 24: Cenário SC3_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,001953	0,003906	0,003906	0,001953	0,003906
TG2	0,001953		1	0,01367	0,1055	0,1309	0,02734
TG3	0,001953	1		0,375	1	1	0,2324
TG4	0,003906	0,01367	0,375		0,625	0,8457	0,1602
TG6	0,003906	0,1055	1	0,625		0,5566	0,9219
TG8	0,001953	0,1309	1	0,8457	0,5566		0,1309
TG10	0,003906	0,02734	0,2324	0,1602	0,9219	0,1309

Figura 57: Cenário SC3_SN - Grafo de ordem parcial.

A.8 Cenário SC4_SN

Figura 58: Cenário SC4_SN - Mapa.

Tabela 25: Cenário SC4_SN - Média e desvio padrão dos confrontos.

TG1 (1) vs (9) TG2	TG1 (0) vs (10) TG3	TG1 (0) vs (10) TG4
70626,5 vs 84209,2 ± 14015,0 vs ± 11545,2	65553,5 vs 82803,9 ± 15535,7 vs ± 15592,1	69916 vs 93577,7 ± 13610,2 vs ± 12931,2
TG1 (1) vs (9) TG6	TG1 (2) vs (8) TG8	TG1 (0) vs (10) TG10
68762,9 vs 91483 ± 10296,9 vs ± 12711,1	68915,1 vs 91306,8 ± 11890,5 vs ± 18578,9	62827,2 vs 78873,7 ± 7121,4 vs ± 13688,7
TG2 (1) vs (9) TG3	TG2 (3) vs (7) TG4	TG2 (1) vs (9) TG6
83991,8 vs 91578,3 ± 10550,8 vs ± 8942,0	78121,2 vs 85994,1 ± 15713,9 vs ± 16251,8	74257,6 vs 82527,2 ± 15917,9 vs ± 15624,2
TG2 (6) vs (4) TG8	TG2 (2) vs (8) TG10	TG3 (4) vs (6) TG4
84448,3 vs 84758,1 ± 16784,3 vs ± 13615,9	74280,7 vs 81918,7 ± 18150,2 vs ± 17223,1	86945,4 vs 88643,7 ± 17396,5 vs ± 15034,0
TG3 (6) vs (4) TG6	TG3 (5) vs (5) TG8	TG3 (6) vs (4) TG10
88562,6 vs 85301,8 ± 17696,3 vs ± 9827,8	86474,5 vs 84759,8 ± 15973,9 vs ± 16878,0	84145,1 vs 82757,2 ± 10481,9 vs ± 12306,9
TG4 (5) vs (5) TG6	TG4 (5) vs (5) TG8	TG4 (9) vs (1) TG10
91281,2 vs 90427,2 ± 19260,5 vs ± 18921,7	82615,1 vs 81391,9 ± 16522,3 vs ± 17778,6	86859,6 vs 81318,2 ± 7002,4 vs ± 8054,2
TG6 (7) vs (3) TG8	TG6 (8) vs (2) TG10	TG8 (4) vs (6) TG10
84789,5 vs 77709,3 ± 12113,2 vs ± 10819,0	82971 vs 74894,4 ± 14747,4 vs ± 15489,0	79744,2 vs 79795,6 ± 15364,5 vs ± 18828,6

Tabela 26: Cenário SC4_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,003906	0,001953	0,001953	0,003906	0,009766	0,08398
TG2	0,003906		0,03711	0,1055	0,009766	0,8457	0,08398
TG3	0,001953	0,03711		0,625	0,625	1	0,7695
TG4	0,001953	0,1055	0,625		0,8457	0,7695	0,01367
TG6	0,003906	0,009766	0,625	0,8457		0,1602	0,04883
TG8	0,009766	0,8457	1	0,7695	0,1602		0,5566
TG10	0,08398	0,08398	0,7695	0,01367	0,04883	0,5566

Figura 59: Cenário SC4_SN - Grafo de ordem parcial.

A.9 Cenário SC4_SS

Figura 60: Cenário SC4_SS - Mapa.

Tabela 27: Cenário SC4_SS - Média e desvio padrão dos confrontos.

TG1 (2) vs (8) TG2		TG1 (2) vs (8) TG3		TG1 (0) vs (10) TG4	
79700,1 ± 4426,5	vs	88495,1 ± 11741,6		73054,1 ± 4680,1	vs 97854,6 ± 12851,5
TG1 (0) vs (10) TG6		TG1 (1) vs (9) TG8		TG1 (2) vs (8) TG10	
73789 ± 8123,3	vs	100042,3 ± 14548,5		68978,9 ± 9304,4	vs 91937,1 ± 22908,6
TG2 (2) vs (8) TG3		TG2 (2) vs (8) TG4		TG2 (3) vs (7) TG6	
90452,2 ± 12829,1	vs	97441,7 ± 15035,7		83612,8 ± 14233,4	vs 91825,8 ± 14076,1
TG2 (4) vs (6) TG8		TG2 (5) vs (5) TG10		TG3 (5) vs (5) TG4	
93260,5 ± 17296,1	vs	98616,2 ± 8184,0		84042,4 ± 9769,4	vs 84072,3 ± 17655,2
TG3 (5) vs (5) TG6		TG3 (5) vs (5) TG8		TG3 (4) vs (6) TG10	
95055,6 ± 13512,9	vs	93119,8 ± 14139,3		82892,6 ± 13849,9	vs 82883,2 ± 21537,6
TG4 (3) vs (7) TG6		TG4 (7) vs (3) TG8		TG4 (7) vs (3) TG10	
91564,2 ± 10836,4	vs	93105,1 ± 8069,0		93446,7 ± 16601,7	vs 79990,9 ± 19239,9
TG6 (5) vs (5) TG8		TG6 (6) vs (4) TG10		TG8 (3) vs (7) TG10	
94170,7 ± 10368,7	vs	91733,6 ± 9240,3		88559,7 ± 16556,0	vs 79492,1 ± 19445,2
				80738,4 ± 11371,2	vs 85270,1 ± 12083,2

Tabela 28: Cenário SC4_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,02734	0,01953	0,001953	0,001953	0,003906	0,01367
TG2	0,02734		0,1934	0,1934	0,2023	0,4922	1
TG3	0,01953	0,1934		0,5566	0,7695	0,5566	1
TG4	0,001953	0,1934	0,5566		0,9219	0,1602	0,02734
TG6	0,001953	0,2023	0,7695	0,9219		0,6953	0,2754
TG8	0,003906	0,4922	0,5566	0,1602	0,6953		0,02734
TG10	0,01367	1	1	0,02734	0,2754	0,02734

Figura 61: Cenário SC4_SS - Grafo de ordem parcial.

A.10 Cenário SC4_NN

Figura 62: Cenário SC4_NN - Mapa.

Tabela 29: Cenário SC4_NN - Média e desvio padrão dos confrontos.

TG1 (1) vs (9) TG2		TG1 (1) vs (9) TG3		TG1 (1) vs (9) TG4	
52203,8 ± 12022,3	vs ± 10118,5	52490,4 ± 9073,2	vs ± 11729,8	45499,8 ± 9201,3	vs ± 11203,5
TG1 (1) vs (9) TG6		TG1 (1) vs (9) TG8		TG1 (1) vs (9) TG10	
52486,3 ± 9466,7	vs ± 7502,0	50593,4 ± 13090,4	vs ± 11938,2	49868,4 ± 8733,1	vs ± 14159,0
TG2 (7) vs (3) TG3		TG2 (4) vs (6) TG4		TG2 (5) vs (5) TG6	
62788,2 ± 16339,1	vs ± 8304,0	59617,8 ± 11215,0	vs ± 11093,3	63382,2 ± 10643,7	vs ± 11209,0
TG2 (8) vs (2) TG8		TG2 (8) vs (2) TG10		TG3 (4) vs (6) TG4	
70436,5 ± 20410,7	vs ± 11549,5	59710,1 ± 9410,5	vs ± 16635,4	58848,1 ± 8398,6	vs ± 12407,1
TG3 (5) vs (5) TG6		TG3 (6) vs (4) TG8		TG3 (5) vs (5) TG10	
62789,1 ± 15344,6	vs ± 15842,7	60783,6 ± 13091,0	vs ± 15089,8	58333 ± 8737,8	vs ± 21448,5
TG4 (4) vs (6) TG6		TG4 (6) vs (4) TG8		TG4 (7) vs (3) TG10	
57749 ± 11558,9	vs ± 11547,7	65148,7 ± 10276,8	vs ± 13770,6	65665,6 ± 13816,3	vs ± 14186,2
TG6 (6) vs (4) TG8		TG6 (7) vs (3) TG10		TG8 (7) vs (3) TG10	
60801,9 ± 9718,4	vs ± 8464,2	65115,1 ± 12870,8	vs ± 20706,9	65822,6 ± 14838,7	vs ± 9388,0

Figura 63: Cenário SC4_NN - Grafo de ordem parcial.

Tabela 30: Cenário SC4_NN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,005859	0,005859	0,003906	0,005859	0,01367	0,02734
TG2	0,005859		0,375	0,4922	0,7695	0,1602	0,4316
TG3	0,005859	0,375		0,625	0,6953	0,4316	0,625
TG4	0,003906	0,4922	0,625		0,9219	0,2754	0,1934
TG6	0,005859	0,7695	0,6953	0,9219		0,4922	0,375
TG8	0,01367	0,1602	0,4316	0,2754	0,4922		0,08398
TG10	0,02734	0,4316	0,625	0,1934	0,375	0,08398	

A.11 Cenário SC5_NN

Figura 64: Cenário SC5_NN - Mapa.

Figura 65: Cenário SC5_NN - Grafo de ordem parcial.

Tabela 31: Cenário SC5_NN - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2		TG1 (0) vs (10) TG3		TG1 (0) vs (10) TG4	
79807,8 ± 3758,2	vs ± 12222,2	105910,8	83309,2 ± 7219,7	116182,8 ± 8356,4	82041,5 ± 6828,4
TG1 (1) vs (9) TG6		TG1 (1) vs (9) TG8		TG1 (1) vs (9) TG10	
80542,9 ± 8440,6	vs ± 9812,5	95753,3	85476,3 ± 10452,4	104629,2 ± 8117,4	77965,8 ± 11897,6
TG2 (5) vs (5) TG3		TG2 (7) vs (3) TG4		TG2 (5) vs (5) TG6	
109646,5 ± 12886,9	vs ± 14042,0	106869,5	109527,4 ± 11771,8	106219,1 ± 15895,2	110682,5 ± 16023,7
TG2 (8) vs (2) TG8		TG2 (5) vs (5) TG10		TG3 (5) vs (5) TG4	
105254,2 ± 15869,7	vs ± 9608,1	94174,5	100289,6 ± 13997,5	97397,1 ± 13366,1	109478,4 ± 16270,8
TG3 (6) vs (4) TG6		TG3 (7) vs (3) TG8		TG3 (8) vs (2) TG10	
104788 ± 10446,5	vs ± 12839,1	101490,2	119661,4 ± 10466,2	107018,9 ± 11597,7	107086,9 ± 10594,5
TG4 (5) vs (5) TG6		TG4 (5) vs (5) TG8		TG4 (7) vs (3) TG10	
105791,6 ± 13209,8	vs ± 14920,6	103500	104001,5 ± 12699,5	105393,2 ± 12552,1	103127 ± 12449,0
TG6 (5) vs (5) TG8		TG6 (7) vs (3) TG10		TG8 (6) vs (4) TG10	
103493,1 ± 12448,9	vs ± 11376,4	107240,2	107779,9 ± 9045,5	100892,2 ± 13972,2	99272,9 ± 20473,6
					95787,1 ± 14382,9

Tabela 32: Cenário SC5_NN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,001953	0,001953	0,01367	0,003906	0,009766
TG2	0,001953		0,6953	0,4922	0,9219	0,02734	0,625
TG3	0,001953	0,6953		0,4922	0,5566	0,08398	0,03711
TG4	0,001953	0,4922	0,4922		0,6953	0,8457	0,1934
TG6	0,01367	0,9219	0,5566	0,6953		0,5566	0,1934
TG8	0,003906	0,02734	0,08398	0,8457	0,5566		0,4316
TG10	0,009766	0,625	0,03711	0,1934	0,1934	0,4316	

A.12 Cenário SC5_SS

Figura 66: Cenário SC5_SS - Mapa.

Tabela 33: Cenário SC5_SS - Média e desvio padrão dos confrontos.

TG1 (1) vs (9) TG2		TG1 (0) vs (10) TG3		TG1 (0) vs (10) TG4	
75905	vs 92074,7	80491,4	vs 99022,2	78752,7	vs 106320,7
± 6846,4	± 14617,2	± 4956,7	± 10528,3	± 4422,2	± 18088,4
TG1 (1) vs (9) TG6		TG1 (1) vs (9) TG8		TG1 (0) vs (10) TG10	
79127,6	vs 95647,2	75121,3	vs 93749,6	71554,1	vs 104118,6
± 5314,0	± 13524,3	± 9731,1	± 12693,7	± 9540,5	± 11225,4
TG2 (4) vs (6) TG3		TG2 (6) vs (4) TG4		TG2 (6) vs (4) TG6	
103222,2	vs 99656,2	97018	vs 96587,6	100491,2	vs 97960,2
± 19664,7	± 18921,6	± 19679,9	± 14377,6	± 15082,0	± 15047,7
TG2 (3) vs (7) TG8		TG2 (3) vs (7) TG10		TG3 (4) vs (6) TG4	
89605,9	vs 96078,3	88248,1	vs 100767,6	103034,9	vs 108803,6
± 18508,1	± 10602,8	± 6840,9	± 14370,0	± 19210,2	± 11826,7
TG3 (4) vs (6) TG6		TG3 (1) vs (9) TG8		TG3 (4) vs (7) TG10	
97730,5	vs 105279,3	88981,4	vs 103910,1	99777,9	vs 113233,9
± 15036,4	± 17230,3	± 18136,2	± 21388,3	± 12563,3	± 18510,5
TG4 (7) vs (3) TG6		TG4 (3) vs (7) TG8		TG4 (4) vs (6) TG10	
108400,6	vs 104458,7	99676	vs 104941,7	97380,1	vs 105399,5
± 10687,8	± 14331,7	± 19146,4	± 22855,1	± 19069,4	± 18002,0
TG6 (6) vs (4) TG8		TG6 (3) vs (7) TG10		TG8 (4) vs (6) TG10	
108624,3	vs 104034	90453	vs 99487,3	94250,3	vs 94531,8
± 16333,2	± 19145,5	± 14148,5	± 22773,7	± 19348,8	± 15021,7

Tabela 34: Cenário SC5_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,005859	0,001953	0,001953	0,005859	0,003906	0,001953
TG2	0,005859		0,6953	0,8457	0,625	0,375	0,04883
TG3	0,001953	0,6953		0,4922	0,4316	0,08398	0,1309
TG4	0,001953	0,8457	0,4922		0,4316	0,4922	0,2754
TG6	0,005859	0,625	0,4316	0,4316		0,6953	0,3223
TG8	0,003906	0,375	0,08398	0,4922	0,6953		1
TG10	0,001953	0,04883	0,1309	0,2754	0,3223	1

Figura 67: Cenário SC5_SS - Grafo de ordem parcial.

A.13 Cenário SC6_SS

Figura 68: Cenário SC6_SS - Mapa.

Tabela 35: Cenário SC6_SS - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2			TG1 (0) vs (10) TG3			TG1 (0) vs (10) TG4		
51225,9 ± 8198,2	vs	66920,2 ± 7486,2	50411,2 ± 9132,8	vs	66936,9 ± 11436,8	53705,5 ± 8621,8	vs	73195,3 ± 10403,6
TG1 (1) vs (9) TG6			TG1 (1) vs (9) TG8			TG1 (4) vs (6) TG10		
45649,4 ± 8279,9	vs	61244,1 ± 8324,3	53525,5 ± 5082,6	vs	65234,2 ± 8509,2	47977,1 ± 9440,6	vs	52247,2 ± 11006,0
TG2 (3) vs (7) TG3			TG2 (3) vs (7) TG4			TG2 (3) vs (7) TG6		
61876,1 ± 8778,5	vs	73715,6 ± 8064,2	60869,2 ± 9718,9	vs	68329,4 ± 6153,8	62828,4 ± 6396,1	vs	70026,9 ± 11279,1
TG2 (5) vs (5) TG8			TG2 (9) vs (1) TG10			TG3 (2) vs (8) TG4		
59673,2 ± 6500,7	vs	59779,7 ± 12451,4	67534 ± 6320,3	vs	55093,4 ± 6658,1	59851,5 ± 12978,6	vs	70044 ± 11264,3
TG3 (7) vs (3) TG6			TG3 (9) vs (1) TG8			TG3 (9) vs (1) TG10		
70256,8 ± 9082,3	vs	63954,8 ± 14341,4	71476,1 ± 8725,6	vs	62809,8 ± 10729,2	66723,8 ± 6248,6	vs	52294,5 ± 13578,9
TG4 (6) vs (4) TG6			TG4 (8) vs (2) TG8			TG4 (8) vs (2) TG10		
65454,2 ± 12607,0	vs	62028,5 ± 12715,2	62144 ± 9819,1	vs	55440,6 ± 13100,9	70043,7 ± 7965,1	vs	55286,8 ± 14482,6
TG6 (8) vs (2) TG8			TG6 (8) vs (2) TG10			TG8 (8) vs (2) TG10		
69298,3 ± 11129,2	vs	59868,8 ± 10211,9	64766,2 ± 9952,9	vs	52589,2 ± 13080,7	59564 ± 13436,7	vs	55307 ± 12509,4

Tabela 36: Cenário SC6_SS - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,001953	0,001953	0,01367	0,003906	0,2754
TG2	0,001953		0,04883	0,06445	0,1309	0,9219	0,005859
TG3	0,001953	0,04883		0,01953	0,2324	0,009766	0,01367
TG4	0,001953	0,06445	0,01953		0,2754	0,1309	0,1309
TG6	0,01367	0,1309	0,2324	0,2754		0,06445	0,02734
TG8	0,003906	0,9219	0,009766	0,1309	0,06445		0,375
TG10	0,2754	0,005859	0,01367	0,1309	0,02734	0,375

Figura 69: Cenário SC6_SS - Grafo de ordem parcial.

A.14 Cenário SC6_SN

Figura 70: Cenário SC6_SN - Mapa.

Tabela 37: Cenário SC6_SN - Média e desvio padrão dos confrontos.

TG1 (0) vs (10) TG2	TG1 (0) vs (10) TG3	TG1 (1) vs (9) TG4
49371,1 vs 65822,1 ± 10278,5 vs ± 10614,2	50443,4 vs 74557,8 ± 9937,5 vs ± 8727,7	51682,1 vs 73748 ± 8306,6 vs ± 11011,9
TG1 (0) vs (10) TG6	TG1 (0) vs (10) TG8	TG1 (4) vs (6) TG10
50954,1 vs 75179,4 ± 7513,9 vs ± 9423,7	49609,3 vs 70040,5 ± 11353,7 vs ± 10800,6	52091,9 vs 61097,3 ± 6001,9 vs ± 15481,7
TG2 (0) vs (10) TG3	TG2 (1) vs (9) TG4	TG2 (2) vs (8) TG6
63004,2 vs 71045,5 ± 7693,9 vs ± 5316,9	58474,6 vs 72509,7 ± 7154,6 vs ± 4946,6	60409,5 vs 67885,4 ± 5220,1 vs ± 8332,6
TG2 (3) vs (7) TG8	TG2 (5) vs (5) TG10	TG3 (3) vs (7) TG4
60031,4 vs 70769 ± 11467,1 vs ± 12827,2	62806,1 vs 55871 ± 5609,1 vs ± 13745,0	68461,3 vs 70745,1 ± 6700,8 vs ± 7090,5
TG3 (4) vs (6) TG6	TG3 (6) vs (4) TG8	TG3 (8) vs (2) TG10
65335,5 vs 70568,6 ± 12429,9 vs ± 11188,1	69367,3 vs 64197,9 ± 7868,5 vs ± 9092,5	67798,8 vs 57893,7 ± 5882,6 vs ± 10447,8
TG4 (3) vs (7) TG6	TG4 (7) vs (3) TG8	TG4 (9) vs (1) TG10
70608,8 vs 75924,5 ± 10394,6 vs ± 7974,0	66271,6 vs 61302,7 ± 9526,4 vs ± 13776,6	70143,4 vs 60870,6 ± 10103,9 vs ± 13661,1
TG6 (6) vs (4) TG8	TG6 (6) vs (4) TG10	TG8 (6) vs (4) TG10
70394,2 vs 65037,8 ± 10447,6 vs ± 5964,4	73963,3 vs 58660,7 ± 13676,0 vs ± 7724,9	67871,3 vs 60603,5 ± 16167,0 vs ± 10329,4

Tabela 38: Cenário SC6_SN - Resultado do teste de “Wilcoxon Pareado” (valor-p).

	TG1	TG2	TG3	TG4	TG6	TG8	TG10
TG1		0,001953	0,001953	0,003906	0,001953	0,001953	0,2754
TG2	0,001953		0,001953	0,003906	0,02734	0,1309	0,2754
TG3	0,001953	0,001953		0,375	0,4316	0,2754	0,08398
TG4	0,003906	0,003906	0,375		0,2754	0,4316	0,03711
TG6	0,001953	0,02734	0,4316	0,2754		0,1055	0,08398
TG8	0,001953	0,1309	0,2754	0,4316	0,1055		0,3223
TG10	0,2754	0,2754	0,08398	0,03711	0,08398	0,3223

Figura 71: Cenário SC6_SN - Grafo de ordem parcial.